



Forms
Contact List
Q&A Instructions
Help Desk

---

**Find Subject Areas:**

**Show Side Menu**      **Search Subject Areas & Legacy Documents:**

---

## Contents: Software Quality Assurance

Effective Date: **November 2002**

Point of Contact: [Software Quality Assurance Subject Matter Expert](#)

---

### Section

### Overview of Content (see section for full process)

#### [Introduction](#)

#### [1. Managing Software Configuration](#)

- Determine the QA Level for the software.
- Manage software configuration applying one of the following subprocesses:
  - Managing Software Configuration QA Level A1/A2
  - Managing Software Configuration QA Level A3

#### [2. Validating and Verifying Software](#)

- Validate and verify software applying one of the following subprocesses:
  - Validating and Verifying Software QA Level A1/A2
  - Validating and Verifying Software QA Level A3

#### [Definitions](#)

#### **Exhibits**

[Sample Acceptance Test Plan](#)

[Sample Disaster Recovery Plan](#)

[Software Quality Assurance Flowchart](#)

#### **Forms**

[Sample Production Approval Form](#)

[Sample Software Change Notice \(SCN\)](#)

[Sample Software Trouble Report \(STR\)](#)

## Training Requirements and Reporting Obligations

This subject area does not contain training requirements.

This subject area does not contain reporting obligations.

## References

[Evaluation of Seller Quality Assurance \(QA\) Programs](#) Subject Area

[Internal Controlled Documents](#) Subject Area

[Purchase Requisition Review for Quality-related Requirements](#) Subject Area

[Training and Qualifications](#) Subject Area

[Work Planning and Control for Experiments and Operations](#) Subject Area

## Standards of Performance

All staff and guests shall comply with applicable Laboratory policies, standards, and procedures, unless a formal variance is obtained.

Managers shall analyze work for hazards, authorize work to proceed, and ensure that work is performed within established controls.

All staff and users shall identify, evaluate, and control hazards in order to ensure that work is conducted safely and in a manner that protects the environment and the public.

## Management System

This subject area belongs to the **Information Resource Management** management system.

[Back to Top](#)

**The only official copy of this file is the one online in SBMS. Before using a printed copy, verify that it is the most current version by checking the document effective date on the BNL SBMS website.**

1.1-012003/standard/3r/3r00t011.htm

Send a question or comment to the [SBMS Help Desk](#)  
[Disclaimer](#)



Forms    Contact List    SBMS Instructions    Help Desk

Find Subject Areas:    Index    Categories    Alpha

Show Side Menu    Search Subject Areas & Legacy Documents:

---

## Introduction: Software Quality Assurance

Effective Date: **November 2002**

Point of Contact: [Software Quality Assurance Subject Matter Expert](#)

---

Software Quality Assurance (SQA) is a systematic effort to prove that a software product is acceptable for BNL use. SQA is based on the premise that software quality must be defined and implemented early in the software development cycle. Consequently, its success is dependent, not only on planning, but on the involvement and support of the administrative and technical staff responsible for defining specifications, design, management, coding, maintenance, and testing of the software.

This subject area focuses on two specific phases in the project life cycle:

1. Managing Software Configuration
2. Validating and Verifying Software

Refer to the [Software Quality Assurance Flowchart](#) for an overview of the procedures described in this subject area.

For the purpose of this subject area, it is assumed that the system design and programming specification phases have been completed. For guidance on the two phases, contact the [Software Quality Assurance Subject Matter Expert](#).

Software Configuration Management is the process used during software implementation to manage the development of computer software products. Verification is defined as the process of determining whether or not the products of a given phase of the implementation cycle fulfill the requirements established during the previous phase (i.e., whether or not it is complete, consistent, and correct enough to support the next phase). Validation is the process of evaluating software throughout its implementation process to ensure compliance with software requirements.

The Software Quality Assurance Subject Area is based on a graded approach. It applies to software that is customized, proposed for use, under development, or being maintained and used, whether that software was developed in-house, licensed from a commercial vendor (Commercial Off-the-Shelf [COTS]) for customized use, obtained from another organization, or otherwise acquired. The subject area includes, but is not limited to the following types of software:

- a. administrative/business-oriented software;
- b. manufacturing-oriented software; and
- c. process control (e.g., Programmable Logic Control instructions).

This subject area does not apply to software used for basic scientific research and development activities unless those activities have environmental, safety, or health impacts.

This subject area does not address the acquisition of software (see the [Evaluation of Seller Quality Assurance \(QA\) Programs](#) and [Purchase Requisition Review for Quality-related Requirements](#) Subject Areas).

[Back to Top](#)

**The only official copy of this file is the one online in SBMS. Before using a printed copy, verify that it is the most current version by checking the document effective date on the BNL SBMS website.**

1.0-112002-/standard/3r/3r00i011.htm

Send a question or comment to the [SBMS Help Desk](#)  
[Disclaimer](#)



Forms
Contact List
SWG Instructions
Help Desk

---

**Find Subject Areas:**

**Show Side Menu**      **Search Subject Areas & Legacy Documents:**

---

*Subject Area: **Software Quality Assurance***

## 1. Managing Software Configuration

Effective Date: **November 2002**

Point of Contact: [Software Quality Assurance Subject Matter Expert](#)

---

## Applicability

This information applies to BNL staff and non-BNL staff who are responsible for managing software configuration.

## Required Procedure

Managing Software Configuration contains three subsections. Follow the steps in [1.1 Determining the QA Level for the Software](#) and choose one of the subsections below when applicable:

[1.2 Managing Software Configuration QA Level A1/A2](#)

[1.3 Managing Software Configuration QA Level A3](#)

**Note:** Staff receiving software (or intending to develop software) that may be classified must notify the [Classified Information Systems Security Site Manager \(ISSM\)](#). Such software may not, under any circumstances, be developed or installed on an unclassified computer.

## 1.1 Determining the QA Level for the Software

Staff perform the following steps when determining the QA Level for the software.

<b>Step 1</b>	<p>Using the criteria in the <a href="#">Screening Guidelines for Work Planning &amp; Control and Application of the Quality Graded Approach</a> exhibit in the <a href="#">Work Planning and Control for Experiments and Operations</a> Subject Area, determine the QA Level for the software.</p> <p><b>Note:</b> This subject area does not apply to</p> <ol style="list-style-type: none"> <li>1. Software used for basic scientific research and development activities unless those activities have environmental, safety, or health impacts;</li> <li>2. Software that is classified QA Level A4.</li> </ol> <p>However, in either case, staff are encouraged to consider all or part of the subject area requirements in meeting their responsibilities to ensure the quality of the software.</p>
---------------	--

## 1.2 Managing Software Configuration QA Level A1/A2

Staff perform the following steps (prior to developing and testing software) when managing software configuration QA Level A1/A2.

<b>Step 1</b>	Obtain written requests and approvals for all new and enhanced software development and file.
<b>Step 2</b>	Isolate the development environment from the production environment (e.g., separate physical device, production system offline, production system with safeguards).
<b>Step 3</b>	Document and review all development/modifications to the source program.
<b>Step 4</b>	Document and file tracking of problems and resolutions.
<b>Step 5</b>	Ensure that source revision control procedures are in place.
<b>Step 6</b>	Ensure that a disaster recovery plan is in place.
<b>Step 7</b>	If a "User's Manual" is required, refer to the section <a href="#">Developing New Controlled Documents</a> in the <a href="#">Internal Controlled Documents</a> Subject Area.
<b>Step 8</b>	If training is required on the software, refer to the section <a href="#">Determining Training and Qualification Requirements</a> in the <a href="#">Training and Qualifications</a> Subject Area.

## 1.3 Managing Software Configuration QA Level A3

Staff perform the following steps (prior to developing and testing software) when managing software configuration QA Level A3.

<b>Step 1</b>	Document all development/modifications to the source program.
<b>Step 2</b>	Ensure that source revision control procedures are in place.
<b>Step 3</b>	Ensure that a disaster recovery plan is in place.
<b>Step 4</b>	If a "User's Manual" is required, refer to the section <a href="#">Developing New Controlled Documents</a> in the <a href="#">Internal Controlled Documents</a> Subject Area.
<b>Step 5</b>	If training is required on the software, refer to the section <a href="#">Determining Training and Qualification Requirements</a> in the <a href="#">Training and Qualifications</a> Subject Area.

## Guidelines

### Software Configuration Management

Software configuration management uses technical and administrative processes to identify, track, and control configuration items and the changes that are made to those items. A configuration item is any software or document component that is designated for configuration management and treated as a single entity in the configuration management process. Staff should perform some of the following typical configuration management activities:

- Identify and use configuration management tools that are compatible with the size and scope of the project.
- Identify and document the functional and physical characteristics of configuration items.
- Establish and maintain baselines--baselined software should undergo approval procedures to authorize

and document changes.

- Establish formal procedures for evaluating and implementing changes (see the [Sample Software Change Notice \(SCN\)](#)).
- Identify change authorities and their responsibilities and determine escalation for problems/decision making.
- Control and track changes to the configuration items (e.g., create an audit trail of each change that is made, name of the person who made the change, why the change was made, the date of the change).
- Record and report change processing and implementation status.
- Assess proposed modifications, enhancements, or additions to determine the effect each change will have on the product.

There are a number of programs that are available to manage software configuration (e.g., Revision Control System [RCS], Clearcase). Contact the [Software Quality Assurance Subject Matter Expert](#) for more information.

Library controls are the procedures and controls, manual or automated, for the handling of source code and object code in their various forms and versions, from the time of their initial approval or acceptance until they have been incorporated into the final deliverable software. These controls should include the following objectives:

- Ensure that different computer program versions are accurately identified and documented.
- Ensure that a consistent software release process is used.
- Ensure that no unauthorized modifications are made to the source code or object programs.
- Ensure that all approved modifications are properly integrated.
- Ensure that the software submitted for testing is the correct version.
- Ensure that infrequently used software is properly archived and stored.

The following is a list of log files that can be kept.

Log	Contents	Read by	Managed by	Entry Mechanism
System Change Log	library changes, configuration file changes, code changes, system administrator changes, application releases, etc.	developers	developers	web form
Application Help Files	program overview documentation which includes a record of application releases	users	developers	file edit (and release)
Trouble Log	record of problems reported to operations support group	diagnostic	diagnostic	web form
System Status	record of controls system components that are out of service, or scheduled to go out of service	users	developer, engineer, technician	web form

### Problem tracking and resolution

Identifying defects (problems) in software is just the beginning of the defect prevention program. To ensure that an effective prevention program is in place, staff should perform the following activities:

- Define and implement a defect data collection process (see the [Sample Software Trouble Report](#)).
- Establish a tracking mechanism, automated or manual, for reporting defects, causes, solutions, and corrective actions. Include measures for identifying the phase in which the defect occurred (e.g., requirements gathering) and the type of defect (e.g., computational or logic error).
- Collect and analyze defect data, search for the root cause of defects, and look for ways to avoid or eliminate similar defects in future projects.

eliminate similar defects in future projects.

## Disaster Recovery

See the [Sample Disaster Recovery Plan](#) exhibit.

## References

[Internal Controlled Documents](#) Subject Area

[Training and Qualifications](#) Subject Area

[Work Planning and Control for Experiments and Operations](#) Subject Area

| [Continue to Next Page](#) |

[Back to Top](#)

**The only official copy of this file is the one online in SBMS. Before using a printed copy, verify that it is the most current version by checking the document effective date on the BNL SBMS website.**

1.0-112002/standard/3r/3r01d011.htm

Send a question or comment to the [SBMS Help Desk](#)  
[Disclaimer](#)



Forms
Contact List
SWG Instructions
Help Desk

---

**Find Subject Areas:**

**Show Side Menu**      **Search Subject Areas & Legacy Documents:**

---

*Subject Area: **Software Quality Assurance***

## 2. Validating and Verifying Software

Effective Date: **November 2002**

Point of Contact: [Software Quality Assurance Subject Matter Expert](#)

---

## Applicability

This information applies to BNL staff and non-BNL staff who are responsible for managing software configuration.

## Required Procedure

For a Commercial Off-the-Shelf (COTS) software purchase from a vendor, the focus will be on testing the vendor's modules to be used in BNL applications for specific functions. The tests will focus on the business model and exercise only those functions and features necessary for BNL to conduct business. The vendor's code will not be specifically tested; it is assumed that the vendor has performed unit and integration testing.

For software that is developed in-house, BNL takes responsibility for the overall system functionality as well as the unit, integration, system and user acceptance testing of all software that has been customized specifically to meet BNL's business requirements.

Validating and Verifying Software contains two subsections:

[2.1 Validating and Verifying Software QA Level A1/A2](#)

[2.2 Validating and Verifying Software QA Level A3](#)

## 2.1 Validating and Verifying Software QA Level A1/A2

Staff perform the following steps when validating and verifying software QA Level A1/A2.

<b>Step 1</b>	Isolate the environment for testing software from the production environment (e.g., separate physical device, production system offline, production system with safeguards).
<b>Step 2</b>	Record and file a test plan that documents input, expected results, and actual results (see the <a href="#">Sample Acceptance Test Plan</a> exhibit).
<b>Step 3</b>	One or more qualified persons (other than the developer if possible) execute the test plan to prove the software satisfies system specifications.
<b>Step 4</b>	Obtain written approval before software is moved into production (see the <a href="#">Sample Production Approval Form</a> ).

## 2.2 Validating and Verifying Software QA Level A3

## 2.2 validating and verifying Software QA Level A3

Staff perform the following steps when validating and verifying software QA Level A3.

<b>Step 1</b>	Test the software to prove it satisfies system specifications.
<b>Step 2</b>	Obtain approval before software is moved into production (see the <a href="#">Sample Production Approval Form</a> ).

### Guidelines

Planning for software testing should start in conjunction with project planning. A project-level software test plan (see the [Sample Acceptance Test Plan](#) exhibit) should be developed for all software products within a software system. This test plan establishes the testing activities necessary to validate that the software requirements have been met and to verify the functionality of the software. The plan also documents a systematic approach to testing throughout the software life cycle.

A comprehensive test plan includes the following types of information:

- Levels of testing (e.g., unit, integration, system, and acceptance);
- Types of tests to be performed (e.g., functional performance, usability, stress, regression, and real-time response);
- Testing strategies (e.g., top down, bottom up, automated, first, beta, black box, white box); and
- Test design (e.g., test cases, fault insertion/error handling, usage scenarios).

[| Go to Previous Page |](#)

[Back to Top](#)

**The only official copy of this file is the one online in SBMS. Before using a printed copy, verify that it is the most current version by checking the document effective date on the BNL SBMS website.**

1.0-112002/standard/3r/3r02d011.htm

Send a question or comment to the [SBMS Help Desk](#)  
[Disclaimer](#)

Empty rectangular box for header information.

*Project Name*

**Sample Acceptance Test Plan**

*Date*

**SAMPLE**  
**TEMPLATE**

## Change Control Page

The following information is being used to control and track modifications made to this document.

- 1) Revision Date:  
Author:  
Section(s):  
Page Number(s):  
Summary of Change(s):

### Title Page

Document Name: *Project Name*  
Acceptance Test Plan

Publication Date: *Month Year*

Project Number: Task: XXXXXXXXXXXXXXXX

Prepared by: XXXX XXXXXX

Approval: \_\_\_\_\_  
*Name and Organization*

Concurrence: \_\_\_\_\_  
*Name and Organization*

**Organizational Title 1**  
**Table of Contents**

Preface .....	ii
1. Project/System Information .....	1-1
1.1 Project Objectives.....	1-1
1.2 System Description.....	1-1
1.3 References .....	1-1
1.4 Outstanding Issues.....	1-1
1.5 Roles and Responsibilities.....	1-1
2. Test Plan.....	2-1
2.1 Scope .....	2-1
2.2 Testing Approach.....	2-1
2.3 Test Schedule.....	2-1
2.4 Problem Reporting and Data Recording .....	2-1
2.5 Resource Requirements .....	2-1
2.6 Test Environment.....	2-2
2.7 Identification of Tests.....	2-2
2.8 Acceptance Test Report.....	2-2
2.9 Corrective Action.....	2-2
3. Test Cases.....	3-1

## **Preface**

---

**Document Version Control:** It is the reader's responsibility to ensure they have the latest version of this document. Questions should be directed to the owner of this document, or the project manager.

**Life Cycle Stage:** *Project Name* is in the Acceptance stage of the software life cycle.

## **1. Project/System Information**

---

### **1.1 Project Objectives**

The business/organizational objectives to be achieved by the project or system.

## 1.2 System Description

Brief description of the system and functions it is intended to perform.

## 1.3 References

Identify sources of information used to develop this document, such as IEEE or project documentation.

## 1.4 Outstanding Issues

Any outstanding issues relative to this project and in particular acceptance testing.

## 1.5 Roles and Responsibilities

The person(s) responsible for, and involved in, all aspects of acceptance testing, both during preparation and execution, and their roles. Document names and roles or reference the Work Breakdown Structure, where these persons and associated activities are typically identified. Include customer/system owner, development/test team, and external persons/groups.

## 2. Test Plan

---

### 2.1 Scope

The scope (boundaries) of the acceptance test - what is, and what is not (that which someone may otherwise believe is) included in the test.

### 2.2 Testing Approach

- X The types of tests which will be performed (e.g., functional testing, security testing, stress testing, timing tests, compliance testing, capacity testing).
- X The level at which testing is performed (e.g., system level, component level, integration level).
- X Test methods, tools, harnesses and procedures to be used.
- X Source(s) of test data.

### 2.3 Test Schedule

A detailed schedule - also known as a Work Breakdown Structure (WBS) - indicating all acceptance testing activities to be performed, and the planned start and end dates for each.

### 2.4 Problem Reporting and Data Recording

- X Description of, or reference to, the problem reporting process which will be used.
- X Description of how test results will be recorded.
- X Description of how problems will be tracked to resolution.

## **2.5 Resource Requirements**

- X Hardware requirements (e.g., hardware items, interfacing equipment).
- X Software requirements (e.g., operating systems, compilers, test drivers, test data generators).
- X Documentation requirements (e.g., test documentation).
- X Staffing requirements (e.g., development team, end users, customers).
- X Test data requirements.
- X Other requirements (e.g., special equipment, rooms, specific times of day or week, turnaround times, training required).

## **2.6 Test Environment**

Describe the plans for setting up the test environment.

## **2.7 Identification of Tests**

List of individual tests and objective(s) of each test. Note: At least one Test Case should be prepared for each test identified here.

## **2.8 Acceptance Test Report**

Document what will be included in the Acceptance Test Report. This report usually closes out the acceptance process.

## **2.9 Corrective Action**

The process that will be used to apply corrections and re-test those cases which fail. This process should be iterative, until each test case has successfully executed.

# **3. Test Cases**

---

The attached Test Scenario Specification Form provides a template for use in preparing individual test cases. The form can be customized as appropriate.

The following is a description of the content of key fields. The others, e.g., Written By, are deemed to be self explanatory.

- Version No.: The version number of the software being tested.
- Release No.: The release number of the software being tested.
- Build #: The tracking number associated with the module, or set of modules, packaged together that perform the function tested by this test case.

Test Scenario#: A number (sequential) assigned to this test case for tracking purposes.

Requirement #: The number of the requirement that will be proven by this test case.

Environment: E.G., mainframe, client/server.

Machine tested: E.G., PC, server.

Retry #: A sequential number representing the number of times the test case has been executed.

STEP#: A number (sequential) associated with the step to be performed in the test scenario.

Description: What happens in this step.

SAMPLE

*[SYSTEM] [Module]*  
**Test Scenario Specification Form**

<b>Version No.:</b>		<b>Build #:</b>		<b>Requirement #:</b>		<b>Page:</b>
<b>Release No.:</b>		<b>Test Scenario #:</b>		<b>Environment:</b>		<b>of</b>
				<b>Machine Tested:</b>		
<b>Written By</b>	<b>Date</b>	<b>Reviewed By</b>	<b>Date</b>	<b>Executed By</b>	<b>Date</b>	<b>Retry #</b>

**Instructions:**

**Test Scenario Objective:**

**Assumptions/Dependencies:**

**Test Files/Test Data:**

SAMPLE

STEP#	DESCRIPTION	EXPECTED RESULT	ACTUAL RESULT

# SAMPLE DISASTER RECOVERY PLAN

FOR

Company

Prepared by:

**SAMPLE**

## Table of Contents<sup>1</sup>

1.0	Plan Introduction	1
1.1	Mission and Objectives	2
1.2	DRP Scope	3
1.3	Authorization	4
1.4	Responsibility	5
1.5	Key Plan Assumptions	6
1.6	Disaster Definition	7
2.0	Business Impact Analysis	8
2.1	Scope	9
2.2	Objectives	10
2.3	Critical Time Frame	11
2.4	Application System Impact Statements	12
	1. Essential	12
	2. Delayed	12
	3. Suspended	12
2.5	Summary Conclusion	13
3.0	Recovery Strategy	14
3.1	Approach	15
3.2	Escalation Plans	16
3.3	Decision Points	17
	PLAN 1	17
	PLAN 2	18
	PLAN 3	18
4.0	Disaster Recovery Organization	20
4.1	Recovery Organization Chart	21
4.2	Disaster Recovery Team	22
4.3	Recovery Team Responsibilities	23
	Recovery Management	23
	Senior Recovery Manager Responsibilities	23
	Pre-Disaster	23
	Post-Disaster	23
	Recovery Manager Responsibilities	24
	Pre-Disaster	24
	Post-Disaster	24

<b>Damage Assessment and Salvage Team</b>	<b>25</b>
Damage Assessment and Salvage Team Responsibilities	25
Pre-Disaster	25
Post-Disaster	25
Physical Security	26
Pre-Disaster	26
Post-Disaster	26
Administration	27
Pre-Disaster	27
Post-Disaster	27
Hardware Installation	28
Pre-Disaster	28
Post-Disaster	28
Systems, Applications and Network Software	29
Pre-Disaster	29
Post-Disaster	29
Communications	30
Pre-Disaster	30
Post-Disaster	30
Operations	31
Pre-Disaster	31
Post-Disaster	31
<b>Disaster Recovery Emergency Procedures</b>	<b>32</b>
5.1 <b>General</b>	<b>34</b>
5.2 <b>Recovery Management</b>	<b>35</b>
5.3 <b>Damage Assessment and Salvage</b>	<b>37</b>
5.4 <b>Physical Security</b>	<b>41</b>
5.5 <b>Administration</b>	<b>43</b>
5.6 <b>Hardware Installation</b>	<b>45</b>
5.7 <b>Systems, Applications &amp; Network Software</b>	<b>47</b>
5.8 <b>Communications</b>	<b>50</b>
5.9 <b>Operations</b>	<b>51</b>
<b>Plan Administration</b>	<b>52</b>
6.1 <b>Disaster Recovery Manager</b>	<b>53</b>
6.2 <b>Distribution of the Disaster Recovery Plan</b>	<b>54</b>
6.3 <b>Maintenance of the Business Impact Analysis</b>	<b>55</b>
6.4 <b>Training of the Disaster Recovery Team</b>	<b>56</b>
6.5 <b>Testing of the Disaster Recovery Plan</b>	<b>57</b>
6.6 <b>Evaluation of the Disaster Recovery Plan Tests</b>	<b>59</b>

<b>6.7</b>	<b>Maintenance of the Disaster Recovery Plan</b>	<b>60</b>
<b>7.0</b>	<b>Appendix</b>	<b>62</b>
	COMPANY Sales Offices	63
	Recovery Team Phone/Address List	64
	Vendor Phone/Address List	65
	Off-Site Inventory	66
	Hardware/Software Inventory	67
	People Interviewed	69
	Preventative Measures	70
	Sample Application Systems Impact Statement	71

## 1.0 Plan Introduction

COMPANY recognizing their operational dependency on computer systems, including the Local Area Network (LAN), Database Servers, Internet, Intranet and e-Mail, and the potential loss of revenue and operational control that may occur in the event of a disaster; authorized the preparation, implementation and maintenance of a comprehensive disaster recover plan.

The intent of a Disaster Recovery Plan is to provide a written and tested plan directing the computer system recovery process in the event of an interruption in continuous service resulting from an unplanned and unexpected disaster.

The Disaster Recovery Plan preparation process includes several major steps as follows:

- ✘ ✘ Identify Systems and Applications currently in use.
- ✘ ✘ Analyze Business Impact of computer impact and determination of critical recovery time frames.
- ✘ ✘ Determine Recovery Strategy
- ✘ ✘ Document Recovery Team Organization
- ✘ ✘ Document Recovery Team Responsibilities
- ✘ ✘ Develop and Document Emergency Procedures
- ✘ ✘ Document Training & Maintenance Procedures.

These steps were conducted and this document represents the completed effort in the preparation of the COMPANY Disaster Recovery Plan.

### 1.5 Key Plan Assumptions

The following assumptions have been established as the basis for the development of the Disaster Recovery Plan:

- ✘ ✘ The plan is designed to recover from the "worst case" destruction of the COMPANY operating environment. The worst case excludes any non-data processing function that may be in close proximity to the data center or workstations.
- ✘ ✘ Although the plan is designed for worst case, inherent in the plan strategy is the ability to recover up to the most minor interruption, which is perhaps a more likely situation.
- ✘ ✘ The plan is base upon a sufficient number of center staff not being

incapacitated to implement and affect recovery. Therefore, the level of detail of the plan is written to a staff experienced in the Company's computer services. Development, testing and implementation of new technologies and applications are suspended so that all resources are available to recover existing critical production processing.

- ✂ ✂ Off-site inventory and equipment acquired through vendors is considered to be the only resource with which to recover computer processing. Items at the original site are not expected to be salvageable and used for recovery. This includes items stored in any on-site security location.

- ✂ ✂ An alternate site (backup computer facility) in which to establish recovery of computer processing is necessary. Time frame requirements to recover computer processing are significantly less than estimated times to repair/reconstruct a data center on an emergency basis.

- ✂ ✂ The computer facilities of the alternative site is not within the scope of this plan and is assumed not to be impacted by any disaster which may interrupt computer operations at COMPANY offices.

## 2.1 Scope

The scope of the Business Impact Analysis is the COMPANY operating departments supported by data center facilities located at \_\_\_\_\_ . This network encompasses the following information technology services:

- ✂ ✂ General business applications, such as word-processing, spreadsheet and database applications

- ✂ ✂ e-Mail

- ✂ ✂ File servers supporting all business operations

- ✂ ✂ Gateway to the host applications and other sites

To determine the maximum time frame allowable, the following COMPANY operating departments were interviewed (See Appendix - People Interviewed):

- ✂ ✂ Information Technology

- ✂ ✂ Sales

- ✂ ✂ Marketing

- ✂ ✂ Credit

- ✂ ✂ Finance

- ✂ ✂ Human Resources

- ✂ ✂ Manufacturing

- ✂ ✂ Distribution

- ✂ ✂ Customer Service

- ✂ ✂ Accounting

- ✂ ✂ Investor Relations

### 4.3.5 Hardware Installation

The Hardware Team is responsible for site preparation, physical planning, and installation of data processing equipment to meet the required processing capacity of COMPANY in the event of a disaster. This includes responsibility for ordering and installing hardware for both the alternative site and the permanent site.

#### Pre-Disaster

- Understand role and responsibilities within the Disaster Recovery Plan

- Work closely with Recovery Management Team to reduce possibility for disaster in data center (See Preventative Measures in Appendix)

- Train employees in emergency preparedness

- Participate in Disaster Recovery Plan tests as required

Maintain current system and LAN configuration in off-site storage

#### Post-Disaster

- Verify with alternative site the pending occupancy requirements

- 

Inspect the alternative site for physical space requirements

- 

Interface with Software, Communications and Operations Team members on space configuration of alternative site

- Coordinate transportation of salvageable equipment to alternative site

- Notify Administration Team of equipment required

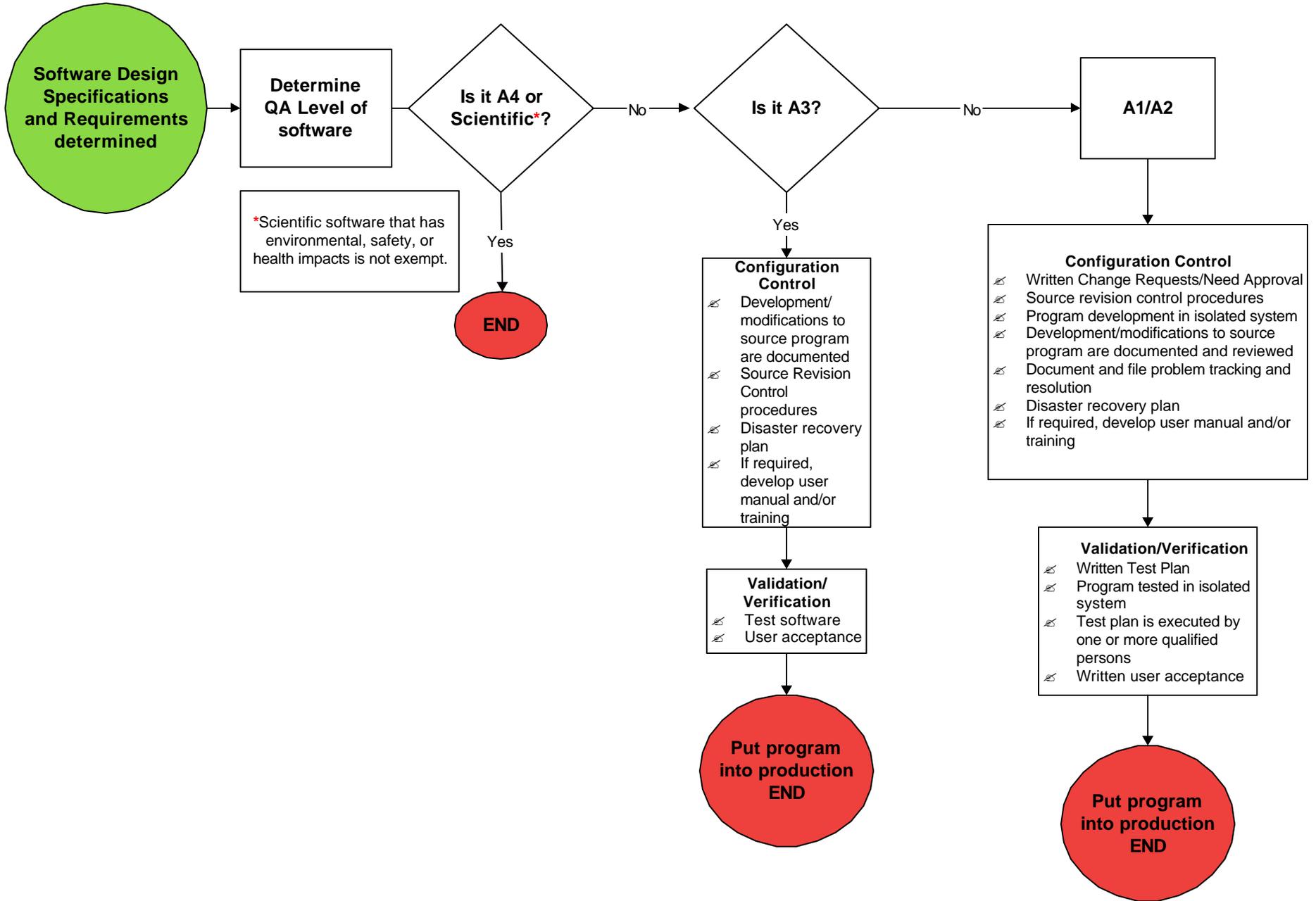
- Ensure installation of \_\_\_ temporary terminals connected to alternative site mainframe

- Plan the hardware installation at the alternative site

- Install hardware at the alternative site

- Plan, coordinate transportation of and install hardware at permanent site, when available

# SOFTWARE QUALITY ASSURANCE FLOWCHART



# SAMPLE PRODUCTION APPROVAL FORM

## Brookhaven National Laboratory QA Implementation

Complete all the following with specific detail. State if Not Applicable.

### Part I

1. Developer:
2. Date Completed by Developer:
3. Date moved into QA: **(Development Representative to fill out)**
4. Date QA Approval or Rejection:
5. Date Moved into Production: **(QA Representative to fill out)**

### Part II

Note: Programmer will create project containing all Objects modified.

6. Project Name:
7. Description of Changes: (Addition, Enhancement, Fix)
8. Modules Effected:
9. Tables, Indexes, Views (Added/Changed/Deleted) Specify if data is to be migrated
10. Panel(s)
11. Panel Groups
12. Menu:
13. Run Control / Process Scheduler to be setup:

# SAMPLE PRODUCTION APPROVAL FORM

## Part III

14. Security Access: (What classes/users it will be assigned to, read only or update)

## Part IV

15. SQR Program:

16. Crystal Reports

17. Operations Instructions:

Specific Directions from Programmer:

Developer Sign Off and Date:

Manager Sign Off and Date:

SAMPLE

# Sample Software Change Notice (SCN)

SCN number: \_\_\_\_\_

Date: \_\_\_\_\_

Originator: \_\_\_\_\_

**Description of change:**

---

---

---

**Reason for Change:**

---

---

**Functions Affected:**

---

---

**Identification of Software Media Affected:**

---

---

**Identification of Documentation Affected:**

---

---

**Software Change Tested/Validated:**     YES     NO

**Software Change Authorization:**

**Date:** \_\_\_\_\_

**Distribution:**

# Software Trouble Report (STR)

STR number: \_\_\_\_\_

Date: \_\_\_\_\_

Originator: \_\_\_\_\_

**Description of Problem:**

---

---

---

**Cause of Problem:**

---

---

**Functions Affected:**

---

---

**Identification of Software Media Affected:**

---

---

**Identification of Documentation Affected:**

---

---

**Corrective Action:**

---

---

---

---

**Software Change Required for Corrective Action:**

YES     NO

**Software Change Authorization:**

**Date:** \_\_\_\_\_

**Distribution:**



Forms
Contact List
SBMS Instructions
Help Desk

**Find Subject Areas:**  Categories

**Show Side Menu**      **Search Subject Areas & Legacy Documents:**

## Definitions: Software Quality Assurance

Effective Date: **November 2002**

Point of Contact: [Software Quality Assurance Subject Matter Expert](#)

Term	Definition
configuration management (software)	A formal engineering discipline that provides the methods and tools to identify and control the software throughout its development and use. Configuration management includes the identification and establishment of baselines; the review, approval, and control of changes; the tracking and reporting of such changes; the audits and reviews of the evolving software product; and the control of interface documentation and project supplier configuration management.
disaster recovery plan	Plan or procedure implemented for recovery from catastrophic system failure.
software	Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. For the purposes of this subject area, "software" includes programs written in compiled or interpreted languages; macros, code, formulas and scripts embedded in COTS such as databases and spreadsheets; Programmable Logic Controller programs, and microprocessor firmware.
software development	The process by which software (or a software module) is created or modified.
software requirement	A condition or capability needed by a customer to solve a problem, achieve an objective, or satisfy a contract/standard/specification (or other formally imposed document).
test plan	A series of inputs, execution instructions, and expected results which are created for the purpose of determining whether software features work correctly or specific requirements have been satisfied.
validation	The process of evaluating software to ensure compliance with software requirements.
verification	A quality management activity that involves checking that the software executes correctly and satisfies the requirements established during the previous phase (i.e., whether or not it is complete, consistent, and correct enough to support the next phase).

[Back to Top](#)

**The only official copy of this file is the one online in SBMS. Before using a printed copy, verify that it is the most current version by checking the document effective date on the BNL SBMS website.**

1.0-112002/standard/3r/3r00l011.htm

Send a question or comment to the [SBMS Help Desk](#)  
[Disclaimer](#)



Forms
Contact List
SBMS Instructions
Help Desk

**Find Subject Areas:**

[Show Side Menu](#)      Search Subject Areas & Legacy Documents:

---

**Revision History: Software Quality Assurance**

 Point of Contact: [Software Quality Assurance Subject Matter Expert](#)


---

## Revision History of this Subject Area

Date	Description	Management System
November 2002	<p>Software Quality Assurance (SQA) is a systematic effort to prove that a software product is acceptable for BNL use. SQA is based on the premise that software quality must be defined and implemented early in the software development cycle. The Software Quality Assurance Subject Area is based on a graded approach. It applies to software that is customized, proposed for use, under development, or being maintained and used, whether that software was developed in-house, licensed from a commercial vendor (Commercial Off-the-Shelf [COTS]) for customized use, obtained from another organization, or otherwise acquired.</p> <p>This subject area replaces the Quality Assurance Manual.</p>	Information Resource Management

[Back to Top](#)

**The only official copy of this file is the one online in SBMS. Before using a printed copy, verify that it is the most current version by checking the document effective date on the BNL SBMS website.**

1.0-112002/standard/3r/3r00a011.htm

Send a question or comment to the [SBMS Help Desk](#)  
[Disclaimer](#)