

7-4.3.2™

IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations

IEEE Power Engineering Society

Sponsored by the
Nuclear Power Engineering Committee



Published by
The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

19 December 2003

Print: SH95168
PDF: SS95168

Recognized as an
American National Standard (ANSI)

IEEE Std 7-4.3.2™-2003
(Revision of
IEEE Std 7-4.3.2-1993)

IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations

Sponsor

Nuclear Power Engineering Committee
of the
IEEE Power Engineering Society

Approved 29 January 2004

American National Standards Institute

Approved 11 September 2003

IEEE-SA Standards Board

Abstract: Additional computer specific requirements to supplement the criteria and requirements of IEEE Std 603™-1998 are specified. Within the context of this standard, the term computer is a system that includes computer hardware, software, firmware, and interfaces. The criteria contained herein, in conjunction with criteria in IEEE Std 603-1998, establish minimum functional and design requirements for computers used as components of a safety system.

Keywords: commercial grade item, diversity, safety systems, software, software tools, software verification and validation

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2003 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 19 December 2003. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

Print: ISBN 0-7381-3810-X SH95168
PDF: ISBN 0-7381-3811-8 SS95168

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS**.”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 7-4.3.2-2003, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations.)

This standard evolved from IEEE Std 7-4.3.2-1993. It represents a continued effort by an IEEE working group to support the specification, design, and implementation of computers in safety systems of nuclear power generating stations.

This standard specifies additional computer-specific requirements (incorporating hardware, software, firmware, and interfaces) to supplement the criteria and requirements of IEEE Std 603-1998. This standard should be used in conjunction with IEEE Std 603-1998 to assure the completeness of the safety system design when a computer is to be used as a component of a safety system.

This standard recognizes that development processes for computer systems continue to evolve. As such, the information presented should not be viewed as the only possible solution. This is in keeping with the desire to use advances in digital technology, provided the criteria and requirements of IEEE Std 603-1998 and this standard are met. For example, while this standard does not address specifically artificial intelligence systems or fourth generation languages, their use is not precluded.

IEEE Std 7-4.3.2-1993 referenced ASME NQA-2a-1990, Part 2.7 (referred to as Part 2.7) to address specific software development requirements. References to ASME NQA-2a-1990 have been removed from this standard, and applicable IEEE standards have been referenced.

This standard does not provide requirements associated with the operation and maintenance of the computer following installation (i.e., surveillance testing frequency). Any problems identified should be addressed through applicable standards that specifically address these requirements.

Clause 5.1 in IEEE Std 603-1998 defines the single-failure criterion. Guidance for the application of this criterion is provided in IEEE Std 379™-1988, Standard Application of the Single Failure Criterion to Nuclear Power Generating Station Class 1E Systems. The approach stated in 5.5 of IEEE Std 379-1988 is also appropriate for potential common-cause failures associated with computer hardware and software that have been developed under the requirements of IEEE Std 603-1998 and this standard. Annex B provides additional guidance for determining the need for design diversity in safety-related computer systems.

The working group revised the guidance in the 1993 standard to further address hazard analysis. These efforts resulted in a complete revision of the existing abnormal conditions and events (ACES) discussion in Annex D. Future work should consider the subject of software safety analysis in addressing system hazards. Additionally, future efforts should consider addressing these topics in the body of the standard.

The Nuclear Regulatory Commission endorsed the concept of requirements grading or classification in SECY-91-192, Digital Computer Systems for Advanced Light Water Reactors. A similar concept of safety classification is presented in ANSI/ANS 51.1983 [B1] and ANSI/ANS 52.1-1983 [B2]. If guidance is provided in a revision to IEEE Std 603-1998, efforts should then be undertaken to apply this concept in a revision to this standard.

This standard does not address justification for the selection of software tools and acceptance criteria for compilers, operating systems, and libraries. The working group felt this subject was outside of the scope of this revision.

During the NPEC preview of this revision of the standard, the topic of safety system software security was discussed. Specifically, the ability of the software system to fulfill its safety related functions in the presence of attacks. Recommendations were made that a future revision of the standard address software risks associated with attacks by insiders and from outside.

The qualification of existing commercial computers, or COTS, is addressed in the body of this standard and in Annex C. It is recognized that COTS dedication is a topic that can be applied to a broader range of Safety Systems equipment than computers, but it is included in this standard because it is not addressed elsewhere at this time. It is recommended (for future IEEE standards work) that a specific standard be developed for COTS dedication. IEEE Std 7-4.3.2 should then be revised to reference the new standard for the COTS dedication process.

In summary, the following major changes were implemented in this version of IEEE Std 7-4.3.2:

- The references were updated to include current IEEE Standards
- The definitions were updated and expanded, and references were provided for definitions obtained from other standards.
- A *Software quality metrics* clause was added. Industry practice is moving towards the use of software quality metrics to assure/monitor/improve software quality in addition to the V&V that has traditionally been applied.
- The *Qualification of existing commercial computers* clause was expanded to provide additional guidance that addresses the move toward the use of more commercial hardware and software in safety systems. This clause was reviewed to ensure consistency with industry guidance (e.g., EPRI guidance).

Discussion during the review led to the action item to move the recommendations and guidance portions of this addition (i.e., “should / may” clauses) to the annex.

- The *Software tools* clause was revised to address expanded use of software tools and methods to confirm suitability (IEC 60880-2, issued last year, specifically addresses the use of software tools.)
- The *Verification and Validation* clause was expanded to support the removal of Annex E, which addressed verification and validation activities. This standard references IEEE Std 1012™, and clarifies requirements that are applicable to safety system software. Whereas IEEE Std 1012 only mentions “independent” V&V in the annex, the authors moved the requirements for IV&V into the body of the standard. Additionally, although different “integrity levels” are defined in IEEE Std 1012, this standard identifies which “integrity level” is applicable to safety system software.
- The *Software configuration management* clause was expanded to provide additional guidance by identifying the key requirements for configuration management for safety system software using the guidance provided in IEEE Std 828™ and IEEE Std 1042™.
- A *Software project risk management* clause was added to provide additional guidance consistent with IEEE Std 1540™ on risk management and IEEE/EIA12207.0 on software life cycle processes.
- A *Fault detection and self-diagnostics* clause was added to address features that are unique to software and computer systems.
- The *Identification* clause has been expanded to include software specific requirements by extending the IEEE Std 603 identification requirements to software.
- Annex A, *Relationship to IEEE Std 603*, was updated to reflect the contents of the current standard.
- Annex B, *Diversity requirements determination*, received minor editorial updates.
- Annex C, *Electromagnetic compatibility*, was deleted because Annex B of IEEE Std 603-1998 addresses the same subject.
- Annex D, *Dedication of existing commercial computers*, was updated to more completely address COTS issues. Additionally, this annex was designated Annex C.
- Annex E, *Verification and validation*, was deleted. V&V requirements were incorporated into the body of the standard. IEEE Std 1012 is referenced in the body to provide guidance.

- Annex F, *Identification and resolution of hazards*, replaced the previous version of this annex. The content was revised to better represent current practices and processes for hazards analysis. Guidance from EPRI [B3] was included in this section. Additionally, this annex was designated as Annex D.^a
- Annex G, *Communication independence*, was designated Annex E.
- Annex H, *Computer reliability*, received minor editorial updates and was designated Annex F.
- Annex G, *Bibliography*, was added to identify informative references.

Participants

This document was prepared by the Application of Programmable Digital Computers to Safety Systems Subcommittee Working Group 6.4 of the IEEE Nuclear Power Engineering Committee. At the time this standard was completed, the Subcommittee Working Group 6.4 had the following membership:

Charles Roslund, Chair
Michael Waterman, Secretary

Larry Erin
Ronald Greenthaler
Myron Hecht

Ronald Jarrett
Ifti Rana
John Scott

James Stewart (deceased)
James Sweeney
John Waclo

At the time this standard was completed, Subcommittee 6 under the Nuclear Power Engineering Committee had the following membership:

Paul Yanosy, Chair
David Horvath, Secretary

Wesley Bowers
Robert Copyak
John Disosway
Britton Grim

Randy Jamison
Jim Keiper
Tom Klein
Glenn Lang
Evangelos Marinos

Mike Miller
Charles Roslund
John Waclo
David Zaprazny

The following members of the balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Stan J. Arnot
Vincent Bacanskas
Farouk Baxter
James Bongarra, Jr.
Wesley Bowers
Daniel Brosnan
John P. Carter
Guru Dutt dhangra
Surin Dureja
Robert Fuld

Wilmer Gangloff
Britton Grim
Randall Groves
David Horvath
Paul Johnson
James T. Keiper
Scott Malcolm
John R. Matras
Richard Meininger
Gary Michel

William Mindick
Radhakrishna Rebbapragada
Charles Roslund
James Ruggieri
Barry Skoras
Neil P. Smith
James Stoner
James Thomas
T.J. Voss
John Waclo

^aThe numbers in brackets correspond to those of the bibliography in Annex G.

When the IEEE-SA Standards Board approved this standard on 11 September 2003, it had the following membership:

Don Wright, *Chair*
Howard M. Frazier, *Vice Chair*
Judith Gorman, *Secretary*

H. Stephen Berger
Joe Bruder
Bob Davis
Richard DeBlasio
Julian Forster*
Toshio Fukuda
Arnold M. Greenspan
Raymond Hapeman

Donald M. Heirman
Laura Hitchcock
Richard H. Hulett
Anant Jain
Lowell G. Johnson
Joseph L. Koepfinger*
Tom McGean
Steve Mills

Daleep C. Mohla
William J. Moylan
Paul Nikolic
Gary Robinson
Malcolm V. Thaden
Geoffrey O. Thompson
Doug Topping
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Alan Cookson, *NIST Representative*
Satish K. Aggarwal, *NRC Representative*

Michelle D. Turner
IEEE Standards Project Editor

Contents

1. Scope	1
2. References	1
3. Definitions and abbreviations	1
3.1 Definitions	1
3.2 Abbreviations and acronyms	5
4. Safety system design basis	6
5. Safety system criteria	6
5.1 Single-failure criterion	6
5.2 Completion of protective action	6
5.3 Quality	6
5.4 Equipment qualification	10
5.5 System integrity	12
5.6 Independence	13
5.7 Capability for test and calibration	14
5.8 Information displays	14
5.9 Control of access	14
5.10 Repair	14
5.11 Identification	14
5.12 Auxiliary features	14
5.13 Multi-unit stations	14
5.14 Human factor considerations	14
5.15 Reliability	15
6. Sense and command features—functional and design requirements	15
7. Execute features—functional and design requirements	15
8. Power source requirements	15
Annex A (informative) Mapping of IEEE Std 603-1998 to IEEE Std 7-4.3.2-2003	16
Annex B (informative) Diversity requirements determination	18
Annex C (informative) Dedication of existing commercial computers	20
Annex D (informative) Identification and resolution of hazards	27
Annex E (informative) Communication independence	39
Annex F (informative) Computer reliability	43
Annex G (informative) Bibliography	48

IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations

1. Scope

This standard serves to amplify criteria in IEEE Std 603™-1998 to address the use of computers as part of safety systems in nuclear power generating stations. The criteria contained herein, in conjunction with criteria in IEEE Std 603-1998, establish minimum functional and design requirements for computers used as components of a safety system.

2. References

This guide shall be used in conjunction with the following standards:

IEEE Std 603-1998, IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations.^{1,2}

IEEE Std 1012™-1998, IEEE Standard for Software Verification and Validation.

IEEE Std 1042™-1987 (R1993), IEEE Guide to Software Configuration Management.³

IEEE/EIA Std 12207.0-1996 IEEE/EIA Standard—Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207), Standard for Information Technology—Software life cycle processes.

3. Definitions and abbreviations

3.1 Definitions

For the purposes of this standard, the following terms and definitions apply. The *Authoritative Dictionary of IEEE Standards Terms* [B5]⁴ should be referenced for terms not defined in this clause.

¹IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://standards.ieee.org/>).

²The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

³IEEE Std 1042-1987 (Reaff 1993) has been withdrawn; however, copies can be obtained from Global Engineering, 15 Inverness Way East, Englewood, CO 80112-5704, USA, tel. (303) 792-2181 (<http://global.ihs.com/>).

⁴The numbers in brackets correspond to those of the bibliography in Annex G.

3.1.1 acceptance testing: (1) Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. *See also:* **qualification testing, system testing.** (2) Formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component.

3.1.2 application software: Software designed to fulfill specific needs of a user; for example, software for navigation, payroll, or process control.

3.1.3 architecture: The organizational structure of a system or component.

3.1.4 commercial grade item: An item that is a) not subject to design or specification requirements unique to nuclear facilities; and b) used in applications other than nuclear facilities; and c) ordered from the manufacturer/supplier on the basis of specifications set forth in the manufacturer's published product description (for example, a catalog).

3.1.5 commercial grade item dedication: A process of evaluating and accepting commercial grade items to obtain adequate confidence of suitability for safety application.

3.1.6 complexity: (1) The degree to which a system or system component has a design or implementation that is difficult to understand and verify. (2) Pertaining to any set of structure-based metrics that measure the attribute in definition (1).

3.1.7 component: One of the parts that make up a system. A component may be hardware or software and may be subdivided into other components.

NOTE—The terms “module,” “component,” and “unit” are often used interchangeably or defined to be supplements of one another in different ways dependent upon the context. The relationship of these terms is not yet standardized.

3.1.8 computer: A functional programmable unit that consists of one or more associated processing units and peripheral equipment, that is controlled by internally stored programs, and that can perform substantial computation, including numerous arithmetic or logic operations, without human intervention.

3.1.9 computer instruction: (1) A statement in a programming language, specifying an operation to be performed by a computer and the addresses or values of the associated operands; for example, Move A to B. (2) Loosely, any executable statement in a computer program.

3.1.10 computer program: A combination of computer instructions and data definitions that enable computer hardware to perform computational or control functions.

3.1.11 computer system: A system containing one or more computers and associated software.

3.1.12 configuration: (1) The arrangement of a computer system or component as defined by the number, nature, and interconnections of its constituent parts. (2) In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product.

3.1.13 configuration control: An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

3.1.14 configuration item: An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process.

3.1.15 configuration management: A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements.

3.1.16 correctness: (1) The degree to which a system or component is free from faults in its specification, design, and implementation. (2) The degree to which software, documentation, or other items meet the specified requirements. (3) The degree to which software, documentation, or other items meet user needs and expectations, whether specified or not.

3.1.17 data: (1) A representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means. (2) Sometimes used as a synonym for documentation.

3.1.18 data structure: A physical or logical relationship among data elements, designed to support specific data manipulation functions.

3.1.19 design: (1) The process of defining the architecture, components, interfaces, and other characteristics of a system or component. (2) The result of the process in definition (1).

3.1.20 document: (1) A medium and the information recorded on it, that generally has permanence and can be read by a person or a machine. Examples in software engineering include project plans, specifications, test plans, user manuals. (2) To create a document as in definition (1). (3) To add comments to a computer program.

3.1.21 documentation: (1) A collection of documents on a given subject. (2) Any written or pictorial information describing, defining, specifying, reporting or certifying activities, requirements, procedures, or results. (3) The process of generating or revising a document. (4) The management of documents, including identification, acquisition, processing, storage, and dissemination.

3.1.22 error: (1) The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. For example, a difference of 30 meters between a computed result and the correct result. (2) An incorrect step, process, or data definition. For example, an incorrect instruction in a computer program. (3) An incorrect result. For example, a computed result of 12 when the correct result is 10. (4) A human action that produces an incorrect result. For example, an incorrect action on the part of a programmer or operator.

NOTE—While all four definitions are commonly used, one distinction assigns definition (1) to the word “error,” definition (2) to the word “fault,” definition (3) to the word “failure,” and definition (4) to the word “mistake.”

3.1.23 execution: The process of carrying out an instruction or the instructions of a computer program by a computer.

3.1.24 failure: The inability of a system or component to perform its required functions within specified performance requirements.

NOTE—The fault tolerance discipline distinguishes between a human action (a mistake), its manifestation (a hardware or software fault), the result of the fault (a failure), and the amount by which the result is incorrect (the error).

3.1.25 fault: (1) A defect in a hardware device or component; for example, a short circuit or broken wire. (2) An incorrect step, process, or data definition in a computer program.

NOTE—This definition is used primarily by the fault tolerance discipline. In common usage, the terms “error” and “bug” are used to express this meaning.

3.1.26 firmware: The combination of a hardware device and computer instructions and data that reside as read-only software on that device.

3.1.27 function: A defined objective or characteristic action of a system or component. For example, a system may have inventory control as its primary function.

3.1.28 functional unit: An entity of hardware, software, or both capable of accomplishing a specified purpose.

3.1.29 hardware: Physical equipment used to process, store, or transmit computer programs or data.

3.1.30 hazard: A condition that is a prerequisite to an accident. Hazards include external events as well as conditions internal to computer hardware or software.

3.1.31 hazard analysis: A process that explores and identifies conditions that are not identified by the normal design review and testing process. The scope of hazard analysis extends beyond plant design basis events by including abnormal events and plant operations with degraded equipment and plant systems. Hazard analysis focuses on system failure mechanisms rather than verifying correct system operation.

3.1.32 implementation: (1) The process of translating a design into hardware components, software components, or both. (2) The result of the process in definition (1).

3.1.33 interface: (1) A shared boundary across which information is passed. (2) A hardware or software component that connects two or more other components for the purpose of passing information from one to the other. (3) To connect two or more components for the purpose of passing information from one to the other. (4) To serve as a connecting or connected component as in definition (2).

3.1.34 module: (1) A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from an assembler, compiler, linkage editor, or executive routine. (2) A logically separable part of a program.

3.1.35 procedure: (1) The course of action taken for the solution of a problem. A course of action taken to perform a given task. (2) A written description of a course of action as in definition (1) for example, a documented test procedure. (3) A portion of a computer program that is named and that performs a specific action.

3.1.36 qualification testing: Testing performed to demonstrate to the acquirer that the software item or system meets its specified requirements.

3.1.37 requirement: (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in definition (1) or definition (2).

3.1.38 requirements specification: A document that specifies the requirements for a system or component. Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.

3.1.39 safety system: A system that is relied upon to remain functional during and following design basis events to ensure: the integrity of the reactor coolant pressure boundary; the capability to shut down the reactor and maintain it in a safe shutdown condition; or the capability to prevent or mitigate the consequences of accidents that could result in potential off-site releases.

3.1.40 software: Computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system.

3.1.41 software maintenance: (1) Modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment. (2) The set of activities that takes place to ensure that software installed for operational use continues to perform as intended and fulfill its intended role in system operation. Software maintenance includes improvements, aid to users, and related activities.

3.1.42 software tools: A computer program used in the development, testing, analysis, or maintenance of a program or its documentation. Examples include comparator, cross-reference generator, decompiler, driver, editor, flowcharter, monitor, test case generator, and timing analyzer.

3.1.43 specification: A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or system, and, often, the procedures for determining whether these provisions have been satisfied.

3.1.44 system: A collection of components organized to accomplish a specific function or a set of functions.

3.1.45 system software: Software designed to facilitate the operation and maintenance of a computer system and its associated programs; for example, operating systems, assemblers, utilities.

3.1.46 system testing: Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

3.1.47 testing: (1) The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component. (2) The process of analyzing a software item to detect the difference between existing and required conditions (that is, bugs) and to evaluate the features of the software items.

3.1.48 test plan: (1) A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. (2) A document that describes the technical and management approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, responsibilities, schedules, and required resources for the testing activity.

3.1.49 validation: The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

3.1.50 verification: (1) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. (2) Formal proof of program correctness.

3.1.51 verification and validation (V&V): The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements.

3.2 Abbreviations and acronyms

The following abbreviations and acronyms appear in this standard:

ATWS	anticipated transient without scram
CFR	Code of Federal Regulations
COTS	commercial off-the-shelf

CRC	cyclic redundancy check
FTA	fault tree analysis
FMEA	failure modes and effects analysis
IV&V	independent verification and validation
MTBF	mean time between failures
PROM	programmable read only memory
QA	quality assurance
RAM	random access memory
SAT	site acceptance testing
SVVP	software verification and validation plan
USNRC	United States Nuclear Regulatory Commission
V&V	verification and validation

4. Safety system design basis

NOTE—See Annex A for more information about the relationship of this standard to IEEE Std 603-1998.

No requirements beyond IEEE Std 603-1998 are necessary (see also Annex B).

5. Safety system criteria

The following subclauses list the safety system criteria in the order they are listed in IEEE Std 603-1998. For some criteria, there are no additional requirements beyond what is stated in IEEE Std 603-1998. For other criteria, additional requirements are described in 5.1 through 5.15.

5.1 Single-failure criterion

No requirements beyond IEEE Std 603-1998 are necessary (see also Annex B).

5.2 Completion of protective action

No requirements beyond IEEE Std 603-1998 are necessary.

5.3 Quality

Hardware quality is addressed in IEEE Std 603-1998. Software quality is addressed in IEEE/EIA Std 12207.0-1996 and supporting standards. Computer development activities shall include the development of computer hardware and software. The integration of the computer hardware and software and the integration of the computer with the safety system shall be addressed in the development process.

A typical computer system development process consists of the following life cycle processes:

- Creating the conceptual design of the system, translation of the concepts into specific system requirements
- Using the requirements to develop a detailed system design
- Implementing the design into hardware and software functions

- Testing the functions to assure the requirements have been correctly implemented
- Installing the system and performing site acceptance testing
- Operating and maintaining the system
- Retiring the system

In addition to the requirements of IEEE Std 603-1998, the following activities necessitate additional requirements that are necessary to meet the quality criterion:

- Software development
- Qualification of existing commercial computers (see 5.4.2)
- Use of software tools
- Verification and validation
- Configuration management
- Risk Management

5.3.1 Software development

Computer software shall be developed, modified, or accepted in accordance with an approved software quality assurance (QA) plan consistent with the requirements of IEEE/EIA 12207.0-1996. The software QA plan shall address all software that is resident on the computer at run time (i.e., application software, network software, interfaces, operating systems, and diagnostics). Guidance for developing software QA plans can be found in IEC 60880 (1986-09) [B4] and IEEE Std 730™-1998 [B8].

5.3.1.1 Software quality metrics

The use of software quality metrics shall be considered throughout the software life cycle to assess whether software quality requirements are being met. When software quality metrics are used, the following life cycle phase characteristics should be considered:

- Correctness/Completeness (Requirements phase)
- Compliance with requirements (Design phase)
- Compliance with design (Implementation phase)
- Functional compliance with requirements (Test and Integration phase)
- On-site functional compliance with requirements (Installation and Checkout phase)
- Performance history (Operation and Maintenance phase)

The basis for the metrics selected to evaluate software quality characteristics should be included in the software development documentation. IEEE Std 1061™-1998 [B11] provides a methodology for the application of software quality metrics.

5.3.2 Software tools

Software tools used to support software development processes and verification and validation (V&V) processes shall be controlled under configuration management.

One or both of the following methods shall be used to confirm the software tools are suitable for use:

- a) A test tool validation program shall be developed to provide confidence that the necessary features of the software tool function as required.

- b) The software tool shall be used in a manner such that defects not detected by the software tool will be detected by V&V activities.

Tool operating experience may be used to provide additional confidence in the suitability of a tool, particularly when evaluating the potential for undetected defects.

5.3.3 Verification and validation

NOTE—See IEEE Std 1012-1998 and IEEE Std 1012a™-1998 [B10] for more information about software V&V.

V&V is an extension of the program management and systems engineering team activities. V&V is used to identify objective data and conclusions (i.e., proactive feedback) about digital system quality, performance, and development process compliance throughout the system life cycle. Feedback consists of anomaly reports, performance improvements, and quality improvements regarding the expected operating conditions across the full spectrum of the system and its interfaces.

V&V processes are used to determine whether the development products of an activity conform to the requirements of that activity, and whether the system performs according to its intended use and user needs. This determination of suitability includes assessment, analysis, evaluation, review, inspection, and testing of products and processes.

This standard adopts the IEEE Std 1012-1998 terminology of process, activity and task, in which software V&V processes are subdivided into activities, which are further subdivided into tasks. The term V&V effort is used to reference this framework of V&V processes, activities, and tasks.

V&V processes shall address the computer hardware and software, integration of the digital system components, and the interaction of the resulting computer system with the nuclear power plant.

The V&V activities and tasks shall include system testing of the final integrated hardware, software, firmware, and interfaces.

The software V&V effort shall be performed in accordance with IEEE Std 1012-1998. The IEEE Std 1012-1998 V&V requirements for the highest integrity level (level 4) apply to systems developed using this standard (i.e., IEEE Std 7-4.3.2™). See IEEE Std 1012-1998 Annex B for a definition of integrity level 4 software.

5.3.4 Independent V&V (IV&V) requirements

The previous section addresses the V&V activities to be performed. This section defines the levels of independence required for the V&V effort. IV&V activities are defined by three parameters: technical independence, managerial independence, and financial independence. These parameters are described in Annex C of IEEE Std 1012-1998.

The development activities and tests shall be verified and validated by individuals or groups with appropriate technical competence, other than those who developed the original design.

Oversight of the IV&V effort shall be vested in an organization separate from the development and program management organizations. The V&V effort shall independently select

- a) The segments of the software and system to be analyzed and tested,
- b) The V&V techniques, and
- c) The technical issues and problems upon which to act.