STANDARDIZATION FOR SAFETY SOFTWARE: CURRENT STATUS AND PERSPECTIVES

JULIO GONZALEZ SANZ
Chairman of AECC's Informatics Section (Spain)

ABSTRACT
During the last years, international, national and professional organizations are producing standards and guides for safety-critical software with different approaches, scopes and structures.

This paper shows relevant efforts in the field. Main concepts and definitions are considered key aspects which provides the differences of focus in every sectorial application. Military and Nuclear are the traditional users and producers of these standards, but other sectors, such as technical, medical, transport, etc., must be to profit with the previous experiences gained and to adopt or to adapt the standards produced up to day in their business.

INTRODUCTION

The very rapid innovation and evolution of the information technology and their applications is based to a great extent on software. The software technology, software products and software services are increasing at high rate of development in comparison with conventional technologies. The logical nature of software and the great expansion of the computer systems industry enables it to be used by a large number of users due to its versatility (GON-90).

Currently, the software industry is facing increased demand for safety software products at different levels of criticality in many different application fields: Industrial process control, aerospace, aeronautics, nuclear, medicine, etc. Reliance on software is not universally acclaimed, and programmable safety systems are currently the source of much controversy. For many experts its increased use in safety critical areas is actually decreasing safety.

The problem with software is that "safe" software is a difficult concept to prove, as it can only be tested against its specification and the purpose for which it was written (BOW-90).

The goal of software safety is to ensure that software executes within a potentially hazardous system without causing or contributing to unacceptable risk of loss such as death, injury, property damage, environmental harm, financial ruin and security leaks (LEV-89).

The use of computer in critical applications creates a new problem with regard to evaluation of a system against very stringent requirements. In order to reduce risk in safety-related software, it is vital that the safety requirements are implemented in the system and can be verified before the system is put into operation.

Software Safety is only recently beginning to be considered a unique and important software quality attribute.

With the increasing number of accidents interest in software safety is rising among regulatory agencies, and new standards for safety-critical software are beginning to appear around the world.

The collection of computer related disasters is realised from a few years ago by (NEW-86) shows the importance of the safety-critical systems based on software.

A number of standardization initiatives concerning safety critical computer systems and software are currently being pursued at international, European and national level, in several fields of application: military, industrial computer systems, aeronautical banking, chemical, insurance, communications, transport, etc.

CONCEPTS AND DEFINITIONS

For a better understanding of the activities and the application field of safety critical software standardization, it is necessary to start from definitions (GON-89). An important problem in the software industry is the communication between different actors in the scope in the context of their environment: suppliers, customers, developers, maintainers, products assurance professionals, safety engineers, managers, etc. This communication problem has been taken in to account by the standardization bodies, and professional organizations, which have a great challenge to provide standards and guidelines for all processes and activities in the software life cycle.
There are many terminologies concerned with software attributes and requirements.

Safety software and related terms are concepts interpreted and defined in different forms depending on the focus of the standard or document analysed.

We can find relevant examples of this analysing some standards work. For ISO TC 176/SC1 (Quality Assurance and Quality Management Vocabulary), safety is a quality characteristic in the same level of the conformity reliability, interchangeability, capability, compatibility and dependability.


In the Boehm's and Mc Call's models of software quality, safety is not a component of these models.

This situation shows that it is necessary to make an harmonization of the safety software related term in order to make possible the communication between the different standards users. At the moment the confusion is served.

To include safety as an software quality characteristic in ISO/IEC JTC 1/DP9126 it reduce the scope of standardization work of ISO/IEC/JTC1/SC7/WG3: Information Systems Engineering / Software Engineering and Management. So, the standards doctrine of JTC1/SC7 explicitly exclude the safety software as an item to be considered in their standardization activities. In this way there are two main focus of standardization: standards for safety software and standards for non-safety software. So, we need to produce two parallel standards frameworks.

To produce standards is an activity slow, expensive and not very well understood and recognised by large amount of informatics practitioners, engineers and managers. There are lots of commonality in safety and non-safety software related standards.

A proposal is to make a general and unique standards framework providing expansion of applicability to the safety aspects in the necessaries cases.

**Accident (or accident state)**

1. A state defined under Accident Conditions or severe Accidents (IAE-88b).
2. An unplanned event or series of events that results in death, injury, illness environmental damage to or loss of equipment or property (IEE-90).
3. An unintended event or sequence of events that causes death, injury, environmental or material damage (MOD-89b).

**Accident conditions**

Deviations from operational states in which the releases of radioactive materials are kept to acceptable limits by appropriate design features. These deviations do not include severe accidents. (IAE-88b).

**Critical attribute**

Is one which, if it got out of control, would threaten the development or functionality of the system. There is always a possibility of any single attribute actually getting out of control (EWI-88).

**Hazard**

1. A condition which lead to an accident (MOD-89a)
2. A set of conditions (i.e., a state) that can lead to an accident given certain environmental conditions (LEV-90).

**Programmable electronic system (PES)**

A system based on one or more central processing units (CPUs), connected to sensors and/or actuators, for the purpose of control, protection or monitoring. (IEC - 89 b) (NOTE - the term PES includes systems incorporating: Microprocessors; Programmable Controllers (PCs); Programmable Logic Controllers (PLCs), and other computer based devices).

**Risk**

1. A measure that combine both the likelihood that a system hazard will cause an accident and the severity of that accident (IEE-90).
2. A measure of the severity and likelihood of an accident (MOD-89a).
3. A function of (1) the likelihood of a hazard occurring, (2) the likelihood that the hazard will lead to an accident, and (3) the worst possible potential loss associated with such an accident (LEV-90).

**Safety**

1. The expectation that a system does not, under defined conditions, lead to a state in which human life, economics or environment are endangered (IEC-89a).
2. The achievement of proper operating conditions, prevention of accidents or mitigation of accident consequences, resulting in protection of site
personnel, the public and the environment from undue radiation hazards (IAE-88a).

(2) Is avoidance of or protection against danger to life and limb, the environment and property arising from failures (EWI-88).

(3) The freedom from those conditions that can cause damage to or loss of equipment or injury or death to human being (MIL-62).

(4) The freedom from unacceptable risks of personal harm (EOQ-89).

(5) The state in which the risk of harm or damage is limited to an acceptable level (ISO-89).

Safety critical function
A function where failure could risk human life (MOD-89a).

Safety critical software
(1) Software, whose use in a system, can result in unacceptable risk. Safety critical software includes software whose operation or failure to operate can lead to a hazardous state, software intended to recover from hazardous states and software intended to mitigate the severity of an accident (IEE-90).

(2) Software used to implement a safety-critical function. (MOD-89a).

Safety related software
Software which ensures that a system does not endanger human life, economics or environment. (IEC-89a).

Safety critical system
Is a system used in an environment that may endanger human life or property (EWI-88).

Safety systems
Systems important to safety, provided to assure the safe shutdown of the reactor or the residual heat removal from the core, or to limit the consequences of anticipated operational occurrences or accident conditions (IAE-88b).

Software safety program
A systematic approach to assuring system safety (IEE-90).

Systems safety
The optimum level of safety subject to resource and operational effectiveness constraints attained by applying engineering and system safety management principles throughout the life cycle of a system. (MIL-62).

IMPORTANCE OF SAFETY CRITICAL SOFTWARE STANDARDIZATION

We consider standardization to be the process of developing guides, rules and conventions for the aspects of safety, and to build and to ensure the safety of the software, with the aim of defining simplifying and determining exactly what activities are associated with these aspects.

A basic objective of safety software standardization is that the follow-up during the whole life-cycle of the software product will give a reasonable assurance that the performance and services given by the product correspond, competitively, to the requirements established.

Safety-Critical Software (SCS) is usually approached by the safety-critical development process. The process is specified by developing standards, while quality assurance auditing determines compliance with the standards. (GON - 89).

Some Governments having into account that the computer-controlled systems constitute one of the fastest growing sectors in the world's advances economies, and that among the many types of application is their use in areas where unreliability or failure could have very serious consequences including danger to life and health - from transport to medicine, power stations to factory production, to say nothing of defense equipment, have taken some action related with the standardization in SCS. So, in UK the Government's Interdepartmental Committee on Software Engineering (ICSE) have prepared a report, "Safe IT" to lead and to coordinate plans and activities related to safety-critical software standards (THY-90).

The Governments are concerned to ensure that safety-critical software as a part of safety-related systems are constructed adequately safe complying satisfactory standards.

But not only the Governments are interested in this field. Users, purchasers, industry, and regulators bodies, privates or publics, have showed from a few years their interest and concern at national and international levels, with circles of, studies and works, pre-standards and standards activities. A sample of it is i.e. EWICS (European Workshop on Industrial Computer Systems) or IEC's technical committees TC 45, TC65.

In order to justify the standardization in this field, several aims have been identified:
To continue the development of European and International Standards.

To provide standards like instruments to facilitate the development of technically sound, feasible and generic software products internationally traded.

To provide references for certification and testing of SCS.

To provide instruments for the users and purchasers in order to ensure their rights on the safety of the products and services controlled by software systems.

SAFETY CRITICAL SOFTWARE STANDARDIZATION STATUS

Various studies have been carried out related to standardization for software in general (NAS, 86), Software engineering (TRI, 84), (THA, 82), (BRA, 84), computers (HEC, 84), software quality assurance (MEE, 84), (GON, 86), (GON, 87a), (GON, 87b), (GON, 89), with different criteria and approaches. On the other hand, new general syntheses must be made periodically, incorporating new versions of already published documents and the latest approved projects, as well as the state of development projects for new standards. It is very important to follow up on the activities of international organizations regarding their cooperation and joint activities, and regarding standards already published by other organizations could be adopted by the former. (eg. The CEN EN-29000 series has adopted the ISO-9000 series).

Knowledge of corresponding standards does not only mean domination of leading technology and know-how, which guarantees and protects the companies' interests in agreements and contracts with their suppliers, by making them comply criteria and structures guaranteed by international consensus when such standards are approved; it also serves to generate products with homogeneous quality in the international market.

Standards can be divided into six classes, according to their scope of application (GON 89):

1 - Company level standards, in which agreement must be reached between the Design and Development, Exploitation, Quality, Reliability and Safety, Sales and Purchasing departments.

2 - Industry level standards, developed by a technical-commercial association. In general, agreement is reached by majority vote of the companies with the association. (IEEE/CS).

3 - Government standards, reflecting many degrees of agreement. In some cases, Governments adopt standards developed by private organizations. In others, governmental departments generate the standards. (e.g. NBS).

4 - Standards by consensus, are developed by representatives of all the sectors interested in the use of the standard. (e.g. ECMA, EWICS).

5 - Multinational standards, are developed or adopted by a group of nations with common interests (CEN/CENELEC, NATO, IAEA).

6 - International Standards, are developed by representatives of different national organizations of standards (ISO/IEC).

There are a multiplicity of standards and guidelines for safety-critical software and assessment frameworks with different structures and terminology. This is a great difficulty to apply to software industry. On the other hand, it is difficult for customers to evaluate the software supplier from the point of view of the management.

We shall examine the standards of the most important organizations, from the point of view of their scope, commercial and industrial impact. At international levels, IEC, ISO and IAEA are the most relevant ones. EWICS is a very important contribution of the European countries to the pre-standardization in this area. Special prestandards efforts come from CEE's Programmes, i.e., AIM'S (Applied Informatics in Medicine), MASQUES Project- Medical Application Software Quality Enhancement by Standards-currently in progress, and representing an effort in a particular sector, where the safety-critical software is a key item in the Information Technology System used in the Health Care domain.

At National levels we have included here the standards bodies that we have considered most relevant.

The Table I shows these organizations, their scope and country

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>ORGANIZATION</th>
<th>SCOPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GERMANY</td>
<td>DIN - VDE</td>
<td>CIVIL</td>
</tr>
<tr>
<td>UK</td>
<td>HSE</td>
<td>CIVIL</td>
</tr>
<tr>
<td></td>
<td>MOD</td>
<td>MILITARY</td>
</tr>
<tr>
<td>USA</td>
<td>ANSI / IEEE</td>
<td>CIVIL</td>
</tr>
<tr>
<td></td>
<td>ANSI / ANS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DOD</td>
<td>MILITARY</td>
</tr>
<tr>
<td>SPAIN</td>
<td>AENOR</td>
<td>CIVIL</td>
</tr>
</tbody>
</table>

TABLE I. COUNTRIES AND STANDARDS BODIES

The Table II shows some of the most relevant standards related with safety-critical software.

87
INTERNATIONAL ACTIVITIES


This international organization has as objective the standardization in the field of Electrotechnic Technology, and 40 nations participate.

IEC is voluntary organization and is an independent organization. The members of the IEC are national committees which represent their respective countries.

In Table III is showed the TCs with concerns in safety-critical software and related areas SCs and WGs, standards, and project and their field of application.

Here are included only the standards and activities of TC 45 and TC 65. TC 56 is most devoted to Reliability.

<table>
<thead>
<tr>
<th>STANDARD</th>
<th>ORIGIN</th>
<th>YEAR</th>
<th>TITLE</th>
<th>APPLICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFISC SSH1-1</td>
<td>USA (AIR FORCE)</td>
<td>1985</td>
<td>SOFTWARE SYSTEM SAFETY HANDBOOK</td>
<td>MILITARY</td>
</tr>
<tr>
<td>MIL-Std-882B</td>
<td>USA (DOD)</td>
<td>March</td>
<td>SYSTEM SAFETY PROGRAM REQUIREMENTS</td>
<td>MILITARY</td>
</tr>
<tr>
<td>NAVORD-OD-44942</td>
<td>USA (NAVAL)</td>
<td>19</td>
<td>SYSTEM SAFETY ENGINEERING GUIDELINES</td>
<td>MILITARY</td>
</tr>
<tr>
<td>DOD-2167A</td>
<td>USA (DOD)</td>
<td>29-Feb</td>
<td>DEFENSE SYSTEM SOFTWARE DEVELOPMENT</td>
<td>MILITARY</td>
</tr>
<tr>
<td>IEC-Pub.880</td>
<td>IEC (CEI)</td>
<td>1966</td>
<td>SOFTWARE FOR COMPUTERS IN THE SAFETY SYSTEMS OF NUCLEAR</td>
<td>NUCLEAR</td>
</tr>
<tr>
<td>IEC-Pub.967</td>
<td>IEC (CEI)</td>
<td>1987</td>
<td>PROGRAMMED DIGITAL COMPUTER IMPORTANT TO SAFETY FOR NUCLEAR</td>
<td>NUCLEAR</td>
</tr>
<tr>
<td>MOD. DEF-STD</td>
<td>UK (MOD)</td>
<td>1969</td>
<td>REQUIREMENTS FOR THE PROCUREMENT OF SAFETY CRITICAL SOFTWARE</td>
<td>MILITARY</td>
</tr>
<tr>
<td>00-55</td>
<td></td>
<td></td>
<td>IN DEFENCE EQUIPMENT</td>
<td></td>
</tr>
<tr>
<td>ANSI/IEEE-ANS</td>
<td>USA (IEEE-ANS)</td>
<td>1982</td>
<td>APPLICATION NUCLEAR FOR PROGRAMMABLE DIGITAL COMPUTER SYSTEMS</td>
<td>NUCLEAR</td>
</tr>
<tr>
<td>7-4-3-2</td>
<td></td>
<td></td>
<td>IN SAFETY SYSTEMS OF NUCLEAR POWER GENERATING STATIONS</td>
<td></td>
</tr>
<tr>
<td>UNE 73-404</td>
<td>SPAIN (AENOR)</td>
<td>1990</td>
<td>GARANTIA DE CALIDAD EN LOS SISTEMAS INFORMATICOS APLICADOS A</td>
<td>NUCLEAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INSTALLACIONES NUCLEARES (QUALITY ASSURANCE IN THE NUCLEAR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INSTALLATIONS INFORMATIC SYSTEMS)</td>
<td></td>
</tr>
</tbody>
</table>

TABLE II. STANDARDS RELATED TO SAFETY-CRITICAL SOFTWARE
<table>
<thead>
<tr>
<th>STANDARD BODY</th>
<th>TC/SC/WG</th>
<th>PROJECT TITLE (IN PROGRESS OR NEW ITEM)</th>
<th>APPLICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEC</td>
<td>TC 45/SC45A/WG 3</td>
<td>* UPDATING IEC PUBLICATION 880</td>
<td>NUCLEAR</td>
</tr>
<tr>
<td>IEC</td>
<td>TC 56/WG 10</td>
<td>* IN PROGRESS: * 56 (SEC.) 224 &quot;SOFTWARE RELIABILITY MANAGEMENT&quot; * 56 (SEC.) 236 &quot;SOFTWARE DEPENDABILITY TESTING&quot;</td>
<td>GENERAL</td>
</tr>
<tr>
<td>IEC</td>
<td>TC 56/WG 10</td>
<td>* NEW ITEMS: * SOFTWARE ANALYSIS REQUIREMENTS * SOFTWARE RELIABILITY AND MAINTAINABILITY MANAGEMENT * SOFTWARE TEST METHODS * SOFTWARE RELIABILITY GROWTH</td>
<td>GENERAL</td>
</tr>
<tr>
<td>IEC</td>
<td>TC 65/SC 65 AMIG 9</td>
<td>* PART 7 - ASSESSMENT OF SYSTEM SAFETY</td>
<td>* INDUSTRIAL-PROCESS/ * MACHINERY CONTROL</td>
</tr>
<tr>
<td>IEC</td>
<td>TC 65/SC 65 AMIG 10</td>
<td>* DRAFT-FUNCTIONAL SAFETY OF PROGRAMMABLE ELECTRONIC SYSTEMS; GENERIC ASPECTS. PART 1: GENERAL REQUIREMENTS (SEP. 1989)</td>
<td>* MEDICINE * INDUSTRIAL-PROCESS/ * MACHINERY CONTROL * TRANSPORT</td>
</tr>
</tbody>
</table>

### Table III. IEC's Technical Committees Related With Safety-Critical Software

**IEC TC 45: Nuclear Instrumentation**

This technical committee have the Sub-Committee 45 A: Reactor Instrumentation, concerned with programmable equipment (mainly microprocessor-base) in nuclear protection systems. The WG3 is the author of the IEC Publication 880: "Software for computers in the safety Systems of nuclear power stations", published in 1986, after many years of work, because of the lack of standards upon which to draw at the time (RAT-89).

In this standard is included a few parts which are more relevant to hardware or system than to software, due to the lack of companion documents on these aspects.

This standard is applicable to highly reliable software required for computers to be used in the safety systems of nuclear power plants for safety functions -Class 1 functions according to (IEC-79). This includes the safety actuation systems, the safety system support features and the protection systems.

Special recommendations are included:

- established criteria as for as they affect the software, taking careful account of the high degree of interdependency between hardware and software;
- a general approach to software development to assure the production of highly reliable software required;
- a general approach to software verification and computer system validation;
- procedures for software maintenance, modification and configuration control.

89
A companion document, addressing the system and hardware aspects which were not covered in the standard is the IEC Publication 987: "Programmed digital computer important to safety for nuclear power stations".

It lays down requirements specific to the hardware of digital systems, specially for redundancy independence and structural distribution.

The structure of the standard is formulated in ten clauses and six appendix (A-F) the clauses are:

1. Scope and object.
2. Terms and definitions.
3. Project structure.
4. Software Requirements.
5. Development (design and coding) of safety system software.
6. Verification.
7. Hardware / software integration
8. Computer system validation
10. Operation.

In order to take into account the technological evolution in software engineering practice, an updating of Publication 880, in the form of a supplement, is in course. Requirements to be considered are:

- Software incorporated in the safety system software.
- Software used for the development of safety system software.
- Automatic analysis tools for software code.
- CAE tools for code, data and logic preparation.
- Functional diversity against common-mode failure.
- Acceptance criteria for programming languages.
- Mathematical based formal specification methods and software prototyping.
- Software for hardware self-supervision.
- Software driver.
- Communication software.

Prerequisites for the use of preexisting software should be elaborated, as well as criteria for the evaluation of development, test and acceptance documentation for the product and its operating experience.

Conformance criteria for the requirements Publication 880 should also be developed and hardware/software integration and maintenance should be complemented for points not already covered. Also security aspects and their implications on safety should be included, (RAT -89).

IEC TC55 (Industrial process measurement and control)

This TC is involved in the preparation of standards for systems and elements used for industrial process measurements and control. The SC65A System Considerations, have several WG's concerned with software and safety. There are:

- **WG 8 : Evaluation of system properties.**
  
  Part 7 is concerned with "Assessment of system safety".

- **WG 9: Safe software.**

  This WG has prepared the Draft titles "Software for Computers in the application of Industrial safety-related systems", (July 1989).

This International Standard has been developed to interpret the basic principles for the design of control systems for safety related software in Programmable Electronic Systems (PES's). The standard covers software in dedicated microprocessors, programmable logic controllers (PLC's), multiprocessor distributed systems, large scale central processor systems and other forms of safety-related PES. The standard also discusses the software principles and requirements and should be read in association with appropriate standards on computer hardware and system integration. This standard establishes no additional functional requirements for safety related systems.

The principles applied in developing safety related software include: Top-Down design methods, modularity, verification of each phase, clear documentation, auditable documents and validation testing.

Clearly is established that the designer will distinguish between reliability and safety. An operationally reliable system is not automatically a safe system. The high reliability is a necessary requirement but is not sufficient to ensure safety.

Reliability requirements are concerned with the continuous provision of the system service. Safety requirements are concerned with making a system does not cause accidents. It shall be transparent from the safety requirements, what actions should be taken, if an unforeseen event in the environment leads to an unsafe state.

- **WG 10: Functional safety of Programmable Electronic Systems (PES).**

  Computer based systems, generically referred to as PES's are increasingly being used for the process and machinery control applications. This has created a demand for their use in safety related applications.
The task of this WG is to prepare an IEC Publication which provides guidelines on programmable electronic systems (PES's) having safety function. WG 9 and WG10 are working closely together.


The guidance in this document is primarily concerned with safety-related systems incorporating PES's or other complex electronic devices. National Committees are requested to indicate whether the WG 10 should concentrate on PES's or cover all types of safety-related systems, irrespective of the technology on which they are based.

There is a great variety of PES applications covering a wide range of complexity, risk and hazard potential. The exact prescription of safety measures will therefore be dependent upon the specific application.

This International Standard (IS) has been developed:

- To provide a basis for carrying out systematically the many activities associated with the safety life cycle of safety-related systems, including design and assessment.
- To provide a guidelines framework sufficiently robust and comprehensive to cater for future developments impacted by the rapidly technological changes.
- To allow future development of this IS to reflect changes in safety integrity criteria and assessment techniques.
- To allow future application IS, dealing with the design and assessment of safety related PES's to be developed.

This standard is concerned with safety to persons and the environment, and concentrates on safety-related systems, it is recognised that the consequence of failure could also have serious economic implications. In such cases the standard may also be applicable when assessing the protection of equipment and product.

Examples of the application sectors coming with the scope of the standard include:

- Process industries (emergency-shutdown systems, fire and gas detection systems, boiler controls).
- Manufacturing industries (industrial robots, machine tools).
- Transportation (railway signalling, braking systems, lifts, ...).
- Medicine (miscellaneous electro-medical apparatus, radiography, ...).

This IS will be composed of several parts. Part 1 is concerned with the General Requirements. Others Parts will provide more detailed guidelines that will include:

- Hazard Analysis.
- Risk assessment.
- System safety elements.
- Safety integrity requirements.
- System integrity levels.
- System validation.
- Retro-fitting.
- Documentation.

The content of this IS-Part 1 is the following:

IEC - FUNCTIONAL SAFETY OF PES

1. Scope.
2. Normative References.
3. Definitions and explanation of terms.
4. System considered.
   4.1. General.
   4.2. Programmable Electronic System (PES).
   4.3. Configuration of Systems.
   4.4. Safety-related systems.
   4.5. Classification of system Configuration.
5. Safety Life Cycle Model.
   5.1. General.
   5.2. Hazard analysis and risk assessment.
   5.2.1. Hazard Analysis.
   5.2.2. Risk assessment.
   5.3. Safety Requirements Specification.
      5.3.1. General.
      5.3.2. Functional Requirements Specification.
      5.3.3. Safety Integrity Requirements Specification.
   5.4. Designation of the safety related System.
   5.5. Design and Implementation.
      5.5.1. General.
      5.5.2. Safety Verification.
   5.6. Safety Validation.
      5.6.1. General.
      5.6.2. Safety Validation Plan.
      5.6.3. Safety Validation Test Specification.
      5.6.4. Safety Validation Analyses.
      5.6.5. Safety Validation Tests.
   5.7. Operation and Maintenance.
   5.8. System Modification (Under consideration).
   5.9. Decommissioning (Under consideration).
   5.10. Retro-fit.
   6.1. General
   6.2. System Safety elements.
6.2.2. System Safety element: Safety related hardware reliability.

7. Assessment general framework.
7.1. General.
7.2. Key Steps.

8. Personal competency.


ANNEXES:
ANNEX A (Normative): Risk and Safety Integrity
ANNEX B (Normative): Considerations underlying the guidance.
ANNEX C (Informative): Examples of different PES's to illustrate embedded and applications software.
ANNEX D (Informative): System live-cycle Model.
ANNEX E (Informative): Case study to illustrate aspects of the safety life-cycle model.
ANNEX G (Informative): General Documentation Requirements.

ISO (International Standards Organization)

ISO TC 176 - Quality Assurance and Quality Management.

This TC have published several standards describing the requirements for a generic quality system for two party contractual situations. The ISO 9001. Quality System - Model for Quality Assurance in Design / Development, Production, Installation and Servicing, provides the requirements for such a system the process of development and maintenance of software is different from that of most other types of industrial products. It is therefore necessary to provide additional guidance for quality systems where software products are involved. So, this TC have prepared the standard ISO-9003:

"Guidelines for the Application of ISO 9001 to Software".

This standard is not explicitly concerned with safety-critical software. However, taken into account that safety is considered by TC 176/SC1 - Vocabulary, as an attribute of quality, this standard can be used like a basis to apply the requirements of a quality system for safety software plus additional requirements for specific field of application: aeronautical, medical devices, nuclear, etc.

This guideline is intended to describe the suggested controls and methods for producing software which meets purchaser's requirements. This is done primarily by preventing nonconformity at all stages from development through to maintenance.

IAEA (International Atomic Energy Agency)

The Agency's plans for establishing safety standards for nuclear power plants, referred to as the NUSS Program, include the development of the three documents: Code of practice, Safety Guides and User's Manuals.

The IAEA's "Manual on Quality for Computer Software", (April 1987), provides guidance in the assurance of quality of specification, design, implementation, maintenance and use of computer software related to items and activities important to safety in nuclear power plants.

This guidance is consistent with, on supplements, the requirements and recommendations of the IAEA code in practice 50 - C - QA: Quality Assurance for Safety in Nuclear Power Plants and related Guides on Quality Assurance for Nuclear Power Plants.

The Manual is intended to be used by all those who, in some way, are involved in software for Safety-related applications for Nuclear Power plants. Including auditors who may be called on to audit management systems and product software.

This Manual is applicable to the following types of software which may directly or indirectly influence the safety of the plant:

- **System** - Includes software which provides services as follows which enables application software to function:
  - input and output management,
  - resource allocation,
  - mathematical and system libraries,
  - data management.

- **Application** - Includes software which contributes towards:
  - operating the plant, (for example software for plant monitoring and control systems, plant protection systems, process diagnostic and operator support systems, etc.),
  - recording and storing data, producing reports, data banks, statistics,
  - modelling and calculating for design, safety and reliability analyses,
  - planning maintenance, fault diagnosis, operational use verification,
  - training plant operators with computer based simulators.

- **Support** - Includes all software used in development, testing and maintenance of the above applications as computer programs.

The Manual covers the development of new software, and the maintenance and use of existing software.
The Table IV shows the matrix between the clauses of ISO 9000-3 and their correspondent ISO-9001 (ISO - 90):

<table>
<thead>
<tr>
<th>ISO 9000-3 Clause</th>
<th>CONTENTS</th>
<th>ISO 9001 Clause</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Introduction</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Scope and field of application</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Scope</td>
<td>1.1</td>
</tr>
<tr>
<td>1.2</td>
<td>Field of application</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Definitions</td>
<td>3</td>
</tr>
<tr>
<td>3.1</td>
<td>General</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Additional definitions</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Quality system - framework</td>
<td>4.1</td>
</tr>
<tr>
<td>4.1</td>
<td>Management responsibility</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>Quality system</td>
<td>4.2</td>
</tr>
<tr>
<td>4.3</td>
<td>Quality plan</td>
<td>4.2</td>
</tr>
<tr>
<td>4.4</td>
<td>Internal quality system audits</td>
<td>4.17</td>
</tr>
<tr>
<td>4.5</td>
<td>Corrective action</td>
<td>4.14</td>
</tr>
<tr>
<td>5</td>
<td>Quality system - life cycle activities</td>
<td>4.3</td>
</tr>
<tr>
<td>5.1</td>
<td>Contract reviews</td>
<td>4.3</td>
</tr>
<tr>
<td>5.2</td>
<td>Purchaser's requirements specification</td>
<td>4.3, 4.4</td>
</tr>
<tr>
<td>5.3</td>
<td>Development planning</td>
<td>4.4</td>
</tr>
<tr>
<td>5.4</td>
<td>Quality planning</td>
<td>4.2</td>
</tr>
<tr>
<td>5.5</td>
<td>Design and implementation</td>
<td>4.4, 4.8, 4.13</td>
</tr>
<tr>
<td>5.6</td>
<td>Testing and validation</td>
<td>4.4, 4.10, 4.13, 4.11</td>
</tr>
<tr>
<td>5.7</td>
<td>Acceptance</td>
<td>4.15, 4.10, 4.13</td>
</tr>
<tr>
<td>5.8</td>
<td>Maintenance</td>
<td>4.19</td>
</tr>
<tr>
<td>6</td>
<td>Quality system - supporting activities</td>
<td>4.4, 4.5, 4.8, 4.12, 4.13</td>
</tr>
<tr>
<td>6.1</td>
<td>Configuration management</td>
<td>4.4, 4.5, 4.8, 4.12, 4.13</td>
</tr>
<tr>
<td>6.2</td>
<td>Document control</td>
<td>4.5</td>
</tr>
<tr>
<td>6.3</td>
<td>Quality records</td>
<td>4.16</td>
</tr>
<tr>
<td>6.4</td>
<td>Measurements</td>
<td>4.20</td>
</tr>
<tr>
<td>6.5</td>
<td>Rules, practices and conventions</td>
<td>4.9, 4.11</td>
</tr>
<tr>
<td>6.6</td>
<td>Tools and techniques</td>
<td>4.9, 4.11</td>
</tr>
<tr>
<td>6.7</td>
<td>Purchasing</td>
<td>4.6</td>
</tr>
<tr>
<td>6.8</td>
<td>Included software product</td>
<td>4.7</td>
</tr>
<tr>
<td>6.9</td>
<td>Training</td>
<td>4.18</td>
</tr>
</tbody>
</table>

**TABLE IV. MATRIX OF CORRELATION BETWEEN ISO-9000-3 AND ISO 9001**

The Table V contents of this document is the following:

1. **INTRODUCTION**
   1.1 Objectives of the Manual
   1.2 Need for Software Quality Assurance
   1.3 Applicability of the Manual
   1.4 Software Quality
   1.5 References

2. **SOFTWARE LIFECYCLE**
   2.1 Requirements Specification
   2.2 Functional Specification
   2.3 Architectural and Detailed Design
   2.4 Coding and Software Implementation
   2.5 Integration Testing and Commissioning
   2.6 Operation, Maintenance and Enhancement

3. **QUALITY ASSURANCE PROGRAMME**
   3.1 General
   3.2 Scope of the Quality Assurance Programme
   3.3 Responsibility
   3.4 QA Programme Establishment
   3.5 Quality Assurance Programme Documentation

4. **ORGANIZATION**
   4.1 Structure
   4.2 Interfaces
   4.3 Duties, Responsibilities, Authority
   4.4 Staffing and Training

5. **SOFTWARE DOCUMENT, CONFIGURATION, MEDIA AND SERVICES CONTROL**
   5.1 Software Control and Computing Facilities
   5.2 Configuration Management
   5.3 Media and Services Control

6. **DESIGN CONTROL**
   6.1 Design Specification
   6.2 Design Verification
   6.3 Validation
   6.4 Tools, Techniques, and Methods
   6.5 Design Changes

7. **PROCUREMENT CONTROL**

8. **TESTING**
   8.1 Test Planning
   8.2 Test Performance
   8.3 Test Reviews
   8.4 Acceptance Testing and Certification

9. **NON-CONFORMANCE CONTROL**

10. **CORRECTIVE ACTION**

11. **RECORDS**

12. **AUDITING**

**TABLE V. IAEA's MANUAL**
EUROPEAN ACTIVITIES

CEN (Commission Europeene pour la Normalisation).

The European Committee for Standardization, responsible for promoting, in EEC and EFTA, the application of ISO standards, has adopted, on 1987, the EN 29000-4 series standards which are identical to the ISO 9000-4 series.

Currently each of the 16 member countries have adopted these EN 29000-4 series standards as national standards. Next adoption will be the ISO 9000-3, related to SQNSQM activities.

The EEC legislation requires that the public sector purchasers make reference to standards when procuring IT systems.

It is well recognized in the industry that whatever results from the current ISO 9000-3 to improve the process the results will be at best methods for raising the confidence in the resulting system in a qualitative manner. Without a proper base for software product measurement throughout the lifecycle, there can be no satisfactory quantitative assessment of quality.

At the moment there are no CEN's standards projects related to safety-critical software. However the EEC (European Economic Community) have provided financial support to prepare special prestandards studies at a generic level, i.e. EWICS or support to prestandards project in a particular application field (i.e. AIM's MASQUES project).

EWICS-TC 7: (European Workshop on Industrial Computer Systems-Committee on Safety, Security and Reliability)

In the 1970's a series of workshops were held at Purdue University, West Lafayette, Indiana (USA). News of the workshops reached Europe, and a group of active university and industry based real time computer system developers formed Purdue Workshops Europe. The CEC took an interest in this group, providing meeting facilities and paying travel and subsistence for Brussels meeting. The name was then changed to the EWICS. The contact with Purdue Workshops was maintained.

EWICS have operated with several TC's. The TC 7, was set up to deal with generic issues in Safety, Security and Reliability of industrial real time computers: Their first meeting was held on 24th June 1974.

The EWICS's pre-standards provides help in writing software which is as error free as possible from the very beginning and which can easily be verified. That ease of verification must be the main principle in constructing safety related software. This documents provides a series of general principles, each expanded by a number of specific recommendations whose adoption will help to ensure that the general principle is satisfied.

In 8 October 1985 was signed a Contract between the European Economic Community (EEC) represented by the Commission of the European Communities (the Commission) and The Safety and Reliability Society Ltd. (SRSL) of UK.

Under the contract, the SRSL is required to carry out a programme of work with the object of stimulating the launching of a series of guidelines in the field of computerised real-time computer systems for standardization at the International European, and National Levels.

The technical work programme addresses four main activity areas relevant to industrial real-time safety related computer systems: System Integrity, Software Quality Assurance and Metrics, Design for System Safety and Reliability and Safety Assessment.

The SRSL agreed with the Commission and the members of the European Workshop on Industrial Computer Systems Technical Committee on Safety, Security and Reliability (EWICS-TC 7) that the work programme could best be carried out through the continuance of EWICS-TC 7.

The set of position papers, from number 1 to 13 has updated and published in two books (RED-88), (RED-89).

The Table VI gives the position papers on pre-standards documents related to safety-critical software. The first document of EWICS-TC 7 (Oct. 1981) is the Position paper No 1: "Development of Safety Related Software". As an example, we shows here their content:

1. Principles for Design of safety Related Software.
   1.1. General.
   1.2. Approach to development.
   1.3. Appropriate programming language and compiler.
   1.4. Verification that recommended criteria have been met.
   1.5. Structure of the Guidelines.

   2.1. Guidelines for design and construction Procedures.
   2.2. Guidelines for the structuring of safety related software.
   2.3. Guidelines concerning self supervision of programs.
   2.4. Guidelines for detailed design and coding.
   2.5. Language: Dependent Guidelines.
3. Safety Related Language, Translator and Linkage
Editor.
3.1. General.
3.2. Error Handling.
3.3. Data and variable Handling.
3.4. Timing aspects.

<table>
<thead>
<tr>
<th>POSITION PAPER NUMBER</th>
<th>TITLE OF EWICS's DELIVERED DOCUMENTS ON SAFETY SOFTWARE</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEVELOPMENT OF SAFETY RELATED SOFTWARE</td>
<td>OCT. 1981</td>
</tr>
<tr>
<td>3</td>
<td>GUIDELINES FOR VERIFICATION AND VALIDATION OF SAFETY RELATED COMPUTER SYSTEMS</td>
<td>JUNE 1983</td>
</tr>
<tr>
<td>4</td>
<td>GUIDELINES FOR DOCUMENTATION OF SAFETY RELATED COMPUTER SYSTEMS</td>
<td>OCT. 1984</td>
</tr>
<tr>
<td>5</td>
<td>TECHNIQUES FOR VERIFICATION AND VALIDATION OF SAFETY RELATED SOFTWARE</td>
<td>JAN. 1985</td>
</tr>
<tr>
<td>7</td>
<td>GUIDELINES FOR THE MAINTENANCE AND MODIFICATION OF SAFETY RELATED COMPUTER SYSTEM</td>
<td>1989</td>
</tr>
<tr>
<td>12</td>
<td>GUIDELINES FOR THE ASSESSMENT OF THE SAFETY AND RELIABILITY OF HIGH INTEGRITY INDUSTRIAL COMPUTER SYSTEM</td>
<td>1989</td>
</tr>
<tr>
<td>13</td>
<td>SAFETY ASSESSMENT AND DESIGN OF INDUSTRIAL COMPUTER SYSTEM TECHNIQUES DIRECTORY</td>
<td>1989</td>
</tr>
</tbody>
</table>

TABLE VI. THE EWICS's POSITION PAPERS RELEVANTS TO SAFETY SOFTWARE

NATIONAL AND OTHERS PROFESSIONALS STANDARDS ACTIVITIES

ANSI (American National Standards Institute)

The ANSI is a federation of standards developing and standars using organizations. It provides coordination aims at avoiding duplication of effort and development of conflicting standards.

It is also the organization through which the U.S. participate in the activities of ISO and IEC Several Key member organizations develop and publish most of the Standards finally approved as American National Standards.

The most significant of these activities related to safety software is the ANS (American Nuclear Society) and IEEE (Institute of Electrical and Electronics Engineers).

IEEE (Institute of Electrical and Electronic Engineers)

In order to ensure that software for use in safety-critical systems will satisfy requirements set by a customer or a licensing body, software quality assurance (SOA) should be applied. SOA is a set of planned and systematic actions necessary to provide adequate confidence that software will satisfy given quality requirements.

There is a general consensus that software quality assurance can affect the overall level of safety properties in the software product. These properties and its contribution to the overall system safety are quality elements to be engineered as well as are any other quality aspects. The term software Quality Assurance Function (SOAF) shall in this context cover any activity carried out in the Quality Assurance Process. (EWI-86).

ANSI / IEEE Std. 730 - 1989

This standard, which title is: "IEEE Standard for Software Quality Assurance Plans", provides uniform minimum acceptance requirements for preparation and contents of SOA plans. This standard applies to the development and maintenance of critical software, which is software whose failure would impact safety or cause large financial or social losses. A companion standard is the ANSI/IEEE Std. 983 - 1986, "IEEE Guide for Software Quality Assurance Planning" (in revision now). This guide describes SOA practices, procedures, conventions, techniques and methodologies in support of ANSI/IEEE 730, to provide recommendations reflecting the current state - of - the -
art in the application of the engineering discipline to the development and maintenance of quality computer software (ANS - 86), (ANS - 89).

The contents of ANSI/IEEE - 730 1989 is given in the Table VII:

TABLE VII. CONTENTS OF ANSI/IEEE 730 - 1989

ANSI/IEEE Std. 1012 - 1986

The standard "IEEE Standard for Software Verification and Validation Plans" provides for critical software, uniform and minimum requirements for the format and content of Software Verification and Validation Plans (SVVPs), defines specific minimum V&V tasks and their required inputs and outputs that shall be included in SVVPs and suggests optional V&V tasks to be used to tailor SVVPs as appropriate for the particular V&V effort.

Additional tasks identified by the user of this standard may be included in the SVVP.

This standard requires that the following be defined for each phase:

1. Verification and Validation tasks
2. Methods and Criteria
3. Inputs and Outputs
4. Schedule
5. Resources
6. Risk and assumptions
7. Roles and responsibilities

The SVVP standard derives its scope from ANSI/IEEE Std. 730 - 1984 (A revision will derive from the 1989 version). The structure of their contents is as follows: (see Table VIII)
Currently a Project of Standard Guidelines associated to Std. 1012 is in progress.

**SOFTWARE SAFETY PLAN DRAFT OUTLINE**

1. Introduction
   1.1 Scope
   1.2 Customizing The Plan
   1.3 Definitions

2. Software Safety Plan (Contents)
   2.1 Safety Planning
   2.2 Limits and Accountability
   2.3 Prerequisites and Constraints
   2.4 Related Standards
   2.5 Definitions
   2.6 Safety Management
     2.6.1 Introduction
     2.6.2 Organization and Responsibilities
     2.6.3 Resources
     2.6.4 Safety Logs
     2.6.5 Documentation
     2.6.6 Configuration Management
     2.6.7 Staff Qualifications
     2.6.8 Certification
     2.6.9 Project Life Cycle
     2.6.10 Tool support and Certification
     2.6.11 Non-Developed Software

3. Software Safety Engineering Practices
   3.1 Introduction
   3.2 Hazards
     3.2.1 Analysis, Human Factors
     3.2.2 Review, Formal review (system specifications review).
   3.3 Requirements
     3.3.1 Analysis
   3.4 Design
     3.4.1 Analysis
   3.5 Implementation
     3.5.1 Analysis-Same as 3.4 but for code
   3.6 Software Change analysis
     3.6.1 Analysis
   3.7 Demonstration of satisfaction of safety requirements
     3.7.1 Identification of safety critical modules
     3.7.2 Identification test cases and or analysis
     3.7.3 Trace to hazard analysis
     3.7.4 Demonstration/documentation of effects of test case execution
   3.8 Post Development
     3.8.1 Software integrity
     3.8.2 Installation
     3.8.3 Operations support
     3.8.4 Monitoring and responsibility (malfunction)
     3.8.5 Maintenance (changes)
     3.8.6 Retirement and notification

Appendix
Specifics on Tool Support; use, products etc.

**IEEE P.1228 (1990 - )**

This Standard project was set up in October 1989 and is planned to finalize for balloting in January 1992. The title of this project is "IEEE Standard for Software Safety Plans", (IEEE - 90).

The initial Structure is as follows: (see Table IX)

**ANSI / IEEE - ANS 7 - 4.3.3 (1982)**

This Standard was co-sponsored by IEEE and ANS. Their title is "Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Power Generating Stations", and was published in 1982.

The criteria established provides a means for promoting safe practices for design and evaluation of safety system performance and reliability. The standard applies to all digital computer systems used in safety systems. It has been developed to amplify IEEE Std. 603 - 1980 - (Standard Criteria for Safety Systems for Nuclear Power Generating Stations) because of the software. The standards establishes no additional functional criteria. The amplification required is regard to the method of designing and qualifying the software and the integration HW / SW. The Table of contents is as follows: (see Table X)

**APPLICATION CRITERIA FOR PROGRAMMABLE DIGITAL COMPUTER SYSTEMS IN SAFETY SYSTEMS OF NUCLEAR POWER GENERATING STATIONS OUTLINE**

1. Scope
2. Definitions
3. Computer System Requirements
   3.1 Hardware Requirements
   3.2 Software Requirements
   3.3 Hardware-Software Integration Requirements
4. Software Development
   4.1 Development Plan
   4.2 Design
   4.3 Implementation
5. Hardware-Software Integration
6. Computer System Validation
7. Verification
   7.1 Organization
   7.2 Review and Audit Procedures
   7.3 Software Test and Analysis

**TABLE IX**

**TABLE X**
AENOR (Asociación Española de Normalización)  
(Spanish Association for Standardization)

AENOR is the Spanish organization for standards, and the representative at European (CEN / CENELEC) and international level (ISO, IEC). Their main members are industrial, commercial and users associations. Also are members professional associations and individual persons. It is a private organization, but have the support of the Ministry of Industry and Energy.

The TC 73 - Energy, have delivered a standard titled “Quality Assurance for Computer Systems Applied to Nuclear Installations” (in Spanish: Aseguramiento de la Calidad para sistemas Informáticos aplicados a instalaciones Nucleares). This standard will be published in the first quarter of 1991.

This standard defines the requirements and recommendations to establish and to perform the Quality Assurance programs applicable to all the phases defined in the adopted life cycle model, such as: specification, design, development, tests, integration, installation, acceptance, operation and maintenance, of the informatic systems related to the nuclear installations safety. However, it would be desirable, the total or partial application of this standard to other informatic systems which can affect to the satisfactory operation of the installation.

This standard is based on the criteria of the 10 - CFR-50 Appendix B (Code of Federal Regulations for Nuclear Energy) which applicability is related to Safety of Nuclear Power Stations. Other input are the documents of the IAEA and IEE/EANS.

In order to apply the Quality Assurance program required by the UNE-73-401 standard to a Informatic System it must be prepared a Quality Assurance activities to perform. In order to prepare this plan it will be used the contents of this standard, and of the UNE-73-401 and 73-402 standard as a baseline, and it will be taken into account the detailed planification, which must be established at the beginning of the informatic system project, this detailed planification will include, classified with respect to the time, and establishing the priority and sequence relations existent between them, the phases and activities of the project, procedures, work instructions preparation, test and inspections to perform, and foresight of needed tools.

The standard specifies the aspects to plan be performing the quality assurance program. These aspects are the following:

* Organisation, including:
  - Definition and documentation of the structure of those organisations which perform activities affecting to the informatic systems quality.
  - Identification of the persons who develop quality assurance functions.
  - Definition of relationships between the different participant organisations.

* Documentation related to the life cycle. This documentation will include at least:
  - Requirement specifications.
  - Design specifications.
  - Integration Plan.
  - Verification and Validation Plan.
  - Verification and Validation report.
  - User documentation
  - Maintenance documentation.

And additionally:
  - Informatic System Development Plan.
  - Configuration Management Plan, and

* Standards, Practices and Conventions including a standards, practices and conventions (to be used in the project) list, and the method to watch over and ensure the verification of the established standards, practices and conventions, by indicating the persons or organisations with the responsibility of its continuous implementation, assessment and maintenance.

* Tools, Technics and Methods, including a list of tools, technics and methods (to be used in each life cycle phase), and describing its objectives and application. It will include too the method to watch over and ensure the use of the established tools, technics and methods.

* Verification and Validation, including the verification activities which will be performed during all the phases of the informatic system life cycle and which will have an extension in accordance with the relevance of this system, and the validation activities which will be performed informatic system and the end of the process, to ensure that this system verifies all the specified requirements.

* Configuration Management, including the set of activities whose objective is to perform the following functions:
  - Configuration Identification.
  - Configuration Changes Control.
  - Configuration State register, and
  - Performance of configuration audits.
Control of marketing. Establishment and documentation of control measures to apply to the process of buying the informatic system, in the related to hardware (and it support) and to software. These control measures will apply to the following aspects:
- Bought in documents control, and
- Control of the bought in hardware.

Control of the process.

Control of documentation. There will be established control measures of the informatic system documentation.

Control of the code. Definition of the control measures applicable to the preparation, review and approval, publication, and distribution and change of the code.

Control of the system. Establishment of control measures to ensure that the storage information is retrievable and that it can not be lost or deteriorate, neither during its daily use nor like a consequence of a catastrophic event. The control measures will contain the following aspects:
- Access authorisation, and
- Protection against damage, deterioration or degradation.

Inspection and supervision. Definition of the inspection and supervision activities to ensure the correct development of the informatic system.

Test-Establishment of a test plan including:
- Planning of the tests.
- Performance of the tests.
- Reviews, and
- Tests during operation.

Nonconformances and corrective actions. Establishment of procedures to detect the nonconformances and to implement the needed corrective actions.

Registers. Preparation and storage of quality assurance registers.

Audits. Preparation of an external and internal audits planned and documented system, including:
- Physical configuration.
- Development, and
- Quality Assurance.

STRUCTURE

1. Responsibilities.
2. Definitions.
3. Quality Assurance Program.
4. Organisation.
5. Documentation.
7. Tools, Techniques and methods.
8. Verification and Validations.
10. Control of marketing.
11. Control of processes.
12. Control of the documentation.
13. Control of the code.
14. Control of the system.
15. Inspection and supervision.
17. Nonconformances and corrective actions.
18. Registers.
19. Audits.
20. Preliminary standard, with the title: "Principles

The owner of the nuclear installation is the responsible person, in accordance with the UNE-73-401 standard, of ensuring that the criteria of the applicable Quality Assurance program is correctly implemented to the informatic system.

The owner can delegate to other organisations or specialists the work of establishing and applying the Quality Assurance program or any part of it, but he will have still the responsibility of its effectiveness. At any case it will can be minor the responsibility of any participant in the related to the quality of the activities which he performs during the development or implementation of the information system, or during the performance of the correspondent part of the quality assurance program.

GERMANY

DIN (Deutsches Institut für Normung e.V.)
(Deutsches Standards Institute)

This Institute have prepared a document, considered like a preliminary standard, with the title: "Principles for Computers in Safety-related System" (DIN-VDE 0801) in (1989). A preliminary standard (PS) is the result of a standardization, which has not yet been published as a standard by the DIN because of certain reservations regarding the contents or because the installation procedure deviates from that established in the standard. No standard draft has been published with respect to this preliminary standard. A PS is not part of the German standards. It is necessary to gain experience in order to inform at the German Electrotechnical Commission in the DIN.

There are not existing European or International Standards for the scope of this PS. As a result of this PS it is intended to bring in suggestions for relevant international and European Standardization at the IEC and
CEN/CENELEC. As soon as a corresponding European Standard or a corresponding harmonization document is under consideration and is adopted by the German Standards, and the VDE regulations, this PS will be withdrawn. The Table XI shows the contents of this PS.

<table>
<thead>
<tr>
<th>CONTENTS (REDUCED)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Scope</td>
</tr>
<tr>
<td>2. Definitions and explanations of terms</td>
</tr>
<tr>
<td>3. Generic safety principles</td>
</tr>
<tr>
<td>4. Causes of errors in computers</td>
</tr>
<tr>
<td>5. Measures taken to avoid errors</td>
</tr>
<tr>
<td>6. Measures taken to control errors</td>
</tr>
<tr>
<td>7. Connection between requirement classes and measures</td>
</tr>
<tr>
<td>Appendix A Measures taken to avoid errors</td>
</tr>
<tr>
<td>Appendix B Measures taken to control errors</td>
</tr>
<tr>
<td>Appendix C Checklists</td>
</tr>
<tr>
<td>Appendix F Operation Checkout</td>
</tr>
<tr>
<td>Appendix S Measures taken to avoid errors during the software development</td>
</tr>
<tr>
<td>S.1 Constructive measures</td>
</tr>
<tr>
<td>S.2 Analytical measures</td>
</tr>
</tbody>
</table>

TABLE XI. PRINCIPLES FOR COMPUTERS IN SAFETY-RELATED SYSTEMS

MILITARY STANDARDS

MOD-UK: (MINISTRY OF DEFENCE, U.K.)

The Ministry of Defence's Safety Critical Software Steering Group has determined that the current approach (for ensuring the safety of defence equipment), which is based on system testing and oversight of the design process will, in the long-term become cumbersome and inefficient for the assurance of the safety of increasingly sophisticated software.

The Steering Group therefore have proposed the adoption of Formal Design Methods to assure correct software, based on rigorous mathematical principles, for the implementation of safety critical computer software. The results of the studies performed are embodied in the drafts of Interim Defence Standards 00-55 and 00-56 (MOD-89 a).

On the other hand, the MOD has a clearly articulated policy to prefer the use of international standards, where applicable, in preference to European, national or defence standards.

DEFENCE STANDARD 00-55

This Interim standard (IS) has been published in September of 1989 by the UK's Ministry of Defence. As Douglas Barrie said "this standard has software developers sitting on the edge of their seats. The standard is intended to bring software developers, screaming and kicking if necessary, into the fold of the rigorously defined and designed life cycle.

The watchword for the standard is reliability. It is aimed at setting down how safety critical software should be developed and validated (BAR-89).

The MOD set up a steering group for the standard early in 1987 and a draft of the resulting proposals were circulating by September 1988. The Standard title is: Requirements for the procurement of Safety Critical Software in "Defence Equipment".

By far the most important talking point of the interim standard is its embracing to formal methods as mandatory.

Main goals of the MOD is to reduce to a minimum the overbudget, systems failure and running late.

This Interim Defence Standard is concerned with safety at every stage of the lifecycle of Safety Critical Software in defence equipment. In this context safety addresses the risk to human life irrespective of the number of potential fatalities.

The procedures and practices included in the standard are to be used by all MOD Authorities involved with the specification, procurement, design, development and certification of Safety Critical Software (SCS) and by In-Service Authorities through all stages of maintenance and modification. It is also to be used by Defence Contracts as required by contract statements of work SCS is considered one element of a family of factors contributing to safety.

There are three components in this Standard: the first contains the safety management procedures which cover planning, controlling and supervising the overall process. The second component contains the software engineering practices which cover the methods and tools for the software design and implementation. The third component contains the stages of the project lifecycle which cover all phases of the definition, development and in-service use of Safety Critical Software.
All software procured for defence projects shall be subject to a Hazard Analysis, and only software by this analysis not to be safety-critical shall be exempt from the requirements of this Standard. Software used to control safety-critical functions in defence system shall be defined by and shown to conform to a formal mathematical specification. The techniques used to identify and develop safety-critical elements in a software system are evolving and this Standard will be revised from time to time.

The table below shows the content of this Standard. (Table XII).

**INTERIM DEFENCE STANDARD 00-55**

**CONTENTS**

Section One. General

0 Introduction
1 Scope
2 Warning
3 Related Documents
4 Definitions

Section Two. Safety Management

5 Introduction
6 Organisations And Responsibilities
7 Safety Planning
8 Code of Design Practice
9 Safety Records Log
10 Documentation
11 Configuration Management
12 Staff Qualifications
13 Certification And Acceptance

Section Three. Software Engineering Practices

14 Introduction
15 Hazard Analysis
16 Safety Integrity Analysis
17 Specification
18 Design
19 Design Reviews
20 Implementation
21 Unacceptable Practices
22 Static Code Analysis
23 Verification Testing
24 Validation Testing
25 Tool Support

Section Four. Project Lifecycle

26 Introduction
27 Concept Formulation
28 Feasibility
29 Project Definition

30 Full Development
31 Production
32 In-Service
33 Disposal

**DOD:USA (Department of Defense)**

In the military standards literature is considered that although software does not have any failures, per se, it does contain error mechanisms that cause hazards. Research and experience have shown that the basic causes of software safety problems are (DOD-83): Specification Error; Design Error; Coding Error; Hardware-Induced Error. Not all software errors result in safety problems, and not all software that functions according to specifications is safe.

The software safety analysis has like general objectives to evaluate the design and to identify potential hazards existing in the system and the relative risks involved. So, it is necessary to identify software conditions that could inhibit desired functions or generated undesired functions.

A good understanding of the design methods in the creation of the system is very helpful, if not essential, in conducting the safety analysis.

The generalized hazard categories can be classified as follows:

- Inherent Hazard
- Timing Hazard
- Induced Hazard
- Latent Hazard

**DOD-STD-2167_A**

This standard is a generic guide for software development. Their title is: "Defense System Software Development". It is not specific for safety-critical software, but contains a requirement for the safety analysis. It is necessary to refer others standards and guidelines to comply and support this requirement.

This standard currently have a large influence in the military area, not only in USA, but is NATO countries.

The description of this requirement is the following: "The contractor shall perform the analysis necessary to ensure that the software requirements, design, and operating procedures minimize the potential for hazardous conditions during the operational mission. Any potential hazardous conditions or operating procedures shall be clearly identified and documented.

**CONCLUSIONS**

Over the last ten years, many efforts have made to develop standards and guidelines related to safety-critical
software (SCS). The special characteristics of SCS processes and products have given rise to the need for generating particular standards and guidelines for the application of general standards, as far as those products are concerned.

- In the civil field and at International level the most relevant efforts in SCS Standardization are coming from IEC TC 65.

- European efforts, specially EWICS-TC7 was set up to propose schemes, principles, procedures and guidelines to European and international bodies.

The adoption of the work of EWICS-TC7 is now widespread in national and international standards organizations. Through its memberships, and their reports, EWICS-TC7 have a large influence in the International Technical Committees on safety-critical software standardization process.

- Other efforts of interest are in progress in UK (BSI), Germany (DIN), Spain (AENOR) and perhaps many others, not well diffused..

- Other efforts at national level in Europe are producing some documents of interest, but are limited to specific industrial fields or constrained to be recognised and applied in a particular country.

- In the military field, different USA standards and guides have served as a fundamental references.

- Also, in the military field the recent contribution of UK’s MoD represents a point of reference for news SCS advanced standards.

**BIBLIOGRAPHY/REFERENCES**

(AFI-85) AFISC SSH 1-1
Software System Safety Handbook
California, 1985.

(AIM-89) AIM
MASQUES-Medical Application Software Quality Enhancement by Standards.

(ANS-82) ANSI/IEEE-ANS
ANSI/IEEE-ANS-7.4.3.2-1982.

(ANS-86a) ANSI/IEEE
"IEEE Guide for Software Quality Assurance Planning".

(ANS-86b) ANSI/IEEE
"IEEE Standard for Software Verification and Validation Plans".

(ANS-89) ANSI/IEEE
"IEEE Standard for Software Quality Assurance Plans".

(BAR-89) BARRIE, D.
"Night is right"
DataLink, Jan. 23, 1989; VNU Bus. Pub. BV.

(BOW-90) BOWMAN, I
Safety at the crossroads

(BRA-84) BRANSTAD M., POWELL B., PATRICIA
Software Engineering Project Standards

(DAN-86) DANIELS, B.K.
Harmonization of Safety Standards for PES
In "Safety and Reliability of Programmable Electronic Systems", ed. B.K. Daniels,

(DAN-89) DANIELS, B.K.
Guidelines, Standards, Tools and Practices for Safe Industrial Computer Systems

(DOD-86a) DOD-STD-2167A
Defense System Software Development
Naval Publ. and Forms Center, 5801 Tabor Ave, Philadelphia, PA 19120.

(DAN-87) DANIELS, B.K.
Second Progress Report and Deliverable Items EWICS TC7 Contract between CEC
and The Safety and Reliability Society Ltd.
Management Report

(DIN-89) DIN
Principles for Computers in safety-related systems
DIN-VDE 0801 NOV. 1989 Preliminary Standards.
(DoD-83)  DoD
DoD - STD - 1679A, Weapon System
Software Development

(EOQ-89)  EOQC-GLOSSARY COMMITTEE
Glossary of terms used in the management of
quality.

(EWI-85a)  EWICS/TC7
Guideline for Verification and Validation of
Safety Related Software
Computers and Standards, V4, n1; 1985
(33-41).

(EWI-85b)  EWICS/TC7
Techniques for Verification and Validation of
Safety Related Software
Computers and Standards, V4, n2; 1985
(101-112).

(EWI-86)  EWICS/TC7
Safety Related Software Measures
EWICS-TC7, Internal Doc., Oct. 30, 1986,
Rev. 2.

(EWI-88a)  EWICS/TC7
Guidelines on Safety Related Measures to be
used in Software Quality Assurance

(EWI-89)  EWICS/TC7
Dependability of Critical Computer
Systems 2

(FDA-89)  FDA
Reviewer Guidance for Computer Controlled
Medical Devices
Rev. proposed by the Medical Device Industry
Computer Software Committee (April 1989).

(GON-86)  GONZALEZ SANZ, J.
Panorama actual des Activités de
Normalisation de Genie et Qualité du Logiciel
Gestion de la Qualité. (AFNOR/AFCE/EOQC, Paris, 24-26

(GON-87a)  GONZALEZ SANZ, J.
Standards for Quality and Quality Assurance
in Software Industries Today
Proc. 31st Ann. Conf. EOQC-87, Munich, 1-

(GON-87b)  GONZALEZ SANZ, J. and BRAVO, C.
Normas y Guías de Aplicación para la
Ingeniería y Calidad del Software
Course Notes. Training Sem. AECC, Madrid,
Dic. 1987 (325 pages).

(GON-88)  GONZALEZ SANZ, J.
Software Quality Terminology
Round Table on Quality Terminology. 32nd

(GON-89)  GONZALEZ SANZ, J.
Recent Advances on International Standards
for Software Quality Management and
Software Quality Assurance Assessment
Prod. 33rd Ann. Conf. EOQC-89, Vienna
1990.

(GON-90a)  GONZALEZ SANZ, J.
Quality-Reliability-Dependability of
Software
1990 (196-209).

(GON-90b)  GONZALEZ SANZ, J.
Quality Assurance and Management for Non-
Deliverable Software
Prod. 8th Int. Conf. ISQA-Jerusalem, Nov.
1990.

(GRU-89)  GRUMAN, G.
Software Safety focus of new British
Standard

(HEC, 84)  HECHT, H.
Computer Standards.

(HSE-87)  HSE (Health and Safety Executive)
Programmable Electronic Systems in Safety
Related Applications
HSE, Her Majesty’s Stationary Office,

(IAE-80)  IAEA (International Atomic Energy Agency)
Safety Guide

(IAE-87)  IAEA
Manual on Quality Assurance for Computer
Software
<table>
<thead>
<tr>
<th>Reference</th>
<th>Details</th>
</tr>
</thead>
</table>
(NAS, 86) NASH S.H. and REDWINE S.T.
A Map of World of Software related
Standard Guidelines and Recommended
Practices.
Proc. Comp. Std. Conf. 1986; May; IEEE
Computer Society Press.

(NEU-90) NEUMANN, P.G.
Illustrative Risks to the Public in the Use of
Computer Systems and Related Technology
A compilation of cases that have appeared in
ACM.SIG SOFT quarterly, Software
Engineering Notes (SEN) from 1986.

Evaluation of Safety Critical Software
Commun. ACM, V33, N6, June 1990 (636-
648).

(RAT-86) RATA, J.M.A.
The role of Standards in Systems Reliability,
Safety and Security
IFIP 5.4 WG meeting, Athens, 14-16 April
1986.

(RAT-89) RATA, J.M.A.
Safety Matters Report
Electricit4 de France, Service IMA.
CLAMART, FRANCE.

(REM-88) REMILL, E.F.
Reliability of Critical Computer Systems,
Part 1

(REM-89) REMILL, E.F.
Reliability of Critical Computer Systems,
Part 2

(THA, 84) THAYER R.
The world of software Engineering
Standards.
III SESAW/IEEE; Oct. 1984; San Francisco,
(pp 149-156).

(THY-90) THYNNE, J.C.J.
A Gov. Consultation Doc. on the Safety of
Computer-Controlled Systems. Dept. of

(TRI, 84) TRIPP L.L.
Annotated Software Engineering Standards
Bibliography.
III SESAW/IEEE; Oct. 1984; San Francisco.
(pp. 140-144).

105