



DPC: 24 / 30485921 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 18 January 2024

Origin: International

Latest date for receipt of comments: 6 March 2024

Project No. 2024/00107

Responsible committee: NCE/8 Instrumentation, Control & Electrical Systems of Nuclear Facilities

Interested committees:

Title: BS EN IEC 60880 Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A function

Please notify the secretary if you are aware of any keywords that might assist in classifying or identifying the standard or if the content of this standard

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

**WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD.
THIS DRAFT IS NOT CURRENT BEYOND 6 March 2024**

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <http://standardsdevelopment.bsigroup.com>

Responsible Committee Secretary: **Mr Andranik Ohanian (BSI)**

Direct tel:

E-mail: andranik.ohanian@bsigroup.com

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/Figure/ Table/Note	Type of comment	Commend (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	

PROJECT NUMBER:

IEC 60880 ED3

DATE OF CIRCULATION:

2024-01-12

CLOSING DATE FOR COMMENTS:

2024-04-05

SUPERSEDES DOCUMENTS:

45A/1437A/RR

IEC SC 45A : INSTRUMENTATION, CONTROL AND ELECTRICAL POWER SYSTEMS OF NUCLEAR FACILITIES

SECRETARIAT:

France

SECRETARY:

Mr Alexander WIGG

OF INTEREST TO THE FOLLOWING COMMITTEES:

PROPOSED HORIZONTAL STANDARD:



Other TC/SCs are requested to indicate their interest, if any, in this CD to the secretary.

FUNCTIONS CONCERNED:

☐ EMC☐ ENVIRONMENT☐ QUALITY ASSURANCE☒ SAFETY

This document is still under study and subject to change. It should not be used for reference purposes.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this document are invited to submit, with their comments, notification of any local regulations or technical reasons that may exist and should be considered should this proposal proceed, recognizing that failure to address such requirements could result in the need for new or existing "In Some Countries" clauses. (See [AC/22/2007](#) or [NEW GUIDANCE DOC](#)).

TITLE:

Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions

NOTE FROM TC/SC OFFICERS:

CD1 prepared for circulation to NCs in accordance with WGA3 discussions and decisions at October 2023 meetings. See decision 45A/2023-12 in 45A/1509/RM.

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope.....	10
2 Normative references	10
3 Terms and definitions	11
4 Symbols and abbreviations.....	17
5 General requirements for software projects	17
5.1 General.....	17
5.2 Software types.....	19
5.3 Software development approach	20
5.4 Software project management.....	21
5.5 Software quality assurance plan	22
5.6 Software configuration management	23
6 Software requirements definition	24
6.1 Specification of software requirements.....	24
6.2 Self-supervision	25
6.3 Periodic testing	25
6.4 Documentation.....	26
7 Software design and implementation	27
7.1 Principles for design and implementation	27
7.1.1 Language independent design and implementation principles	27
7.1.2 Design and implementation of software in general-purpose languages	28
7.1.3 Design and implementation of software in application-oriented languages.....	28
7.1.4 Configuration of pre-developed software.....	29
7.1.5 Management of Configuration data	29
7.2 Selection of languages.....	30
7.2.1 General requirements	30
7.2.2 General purpose languages	31
7.2.3 Application-oriented languages and associated automated code generation	32
7.3 Detailed requirements and recommendations.....	32
7.4 Documentation.....	34
8 Software verification	34
8.1 Software verification process	34
8.2 Software verification activities.....	35
8.2.1 Software verification plan	35
8.2.2 Design verification	37
8.2.3 Implementation verification	38
9 Software aspects of system integration.....	40
9.1 General.....	40
9.2 Software integration plan	41
9.3 System integration	41
9.4 Integrated system verification	42
9.5 Fault resolution procedures.....	42
9.6 Software integration report.....	

10	Software aspects of system validation	42
10.1	Software validation plan	42
10.2	System validation	42
10.3	Fault resolution procedures	43
10.4	Software validation report	43
11	Software aspects of system installation	43
11.1	On-site installation of the software	43
11.2	Adaptation of the software to on-site conditions	43
12	Software modification	43
12.1	General	43
12.2	Software modification request	44
12.3	Procedure for executing a software modification	45
12.4	Anomaly report	46
13	Defences against common cause failure due to software	47
14	Software tools for the development of software	47
14.1	General	47
14.2	Selection of software tools	48
14.3	Requirements for software tools	48
14.3.1	Software engineering environment	48
14.3.2	Tool qualification	50
14.3.3	Tool configuration management	50
14.3.4	Translators/compilers	51
14.3.5	Automation of testing	52
15	Qualification of pre-developed software	52
15.1	General	52
15.2	PDS qualification process	55
15.2.1	Evaluation of the functional and performance features	55
15.2.2	Quality evaluation	56
15.2.3	Evaluation of operating experience	58
15.2.4	Comprehensive assessment	60
15.2.5	Maintenance	60
Annex A (normative)	Detailed requirements and recommendations for design and implementation	61
A.1	Design process	61
A.2	Software structure	63
A.3	Self-supervision	67
A.4	Detailed design and coding	68
A.5	Language dependent recommendations	71
Annex B (informative)	Example of application oriented software engineering (software development with application-oriented language)	74
B.1	General	74
B.2	Principle of application of requirements	74
B.3	Application of requirements for software life cycle	75
B.4	Application of requirements for automated code generation	75
B.4.1	General	75
B.4.2	Tools for design and project management	75
B.4.3	Tools for automated generation	76

Annex C (informative) Software verification and testing	78
C.1 Software verification and testing activities	78
C.2 Systematic approaches.....	78
C.3 Statistical approaches	79
C.4 Verification and testing methods.....	81
C.4.1 Selected verification methods	81
C.4.2 Testing methods	83
Annex D (informative) Typical list of software documentation.....	86
Annex E (informative) Tools for production and checking of specification, design and implementation	87
E.1 General	87
E.2 Constructive tools.....	87
E.2.1 Text editor	87
E.2.2 Graphical interface.....	87
E.2.3 Automated code generator.....	87
E.2.4 Proof obligation generator.....	87
E.3 Analytical tools	88
E.3.1 Syntax checker	88
E.3.2 Semantic checker	88
E.3.3 Formal proof generator	88
E.3.4 Animator	88
E.3.5 Compliance analyser	88
Annex F (informative) Correspondence between IEC 61513 and this standard.....	89
Annex G (informative) Relation of this standard with IEC 61508.....	90
G.1 General	90
G.2 Comparison of scope and concepts	90
G.3 Correspondence between this standard and IEC 61508-3:2010	90
 Figure 1 – Activities of the system safety lifecycle (as defined by IEC 61513).....	18
Figure 2 – Software related activities in the system safety lifecycle.....	19
Figure 3 – Development activities of the IEC 60880 software safety lifecycle	21
Figure 4 – Outline of the qualification process of pre-developed software	54
Figure 5 – Life cycle for application oriented software engineering.....	77
 Table 1 – Process and product aspects of design and implementation	33
Table 2 – Correspondence between subclauses of IEC 61513:202X and this standard	89
Table 3 – Correspondence between this standard and IEC 61508-3:2010.....	90

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**NUCLEAR POWER PLANTS –
INSTRUMENTATION AND CONTROL SYSTEMS IMPORTANT TO SAFETY –
SOFTWARE ASPECTS FOR COMPUTER-BASED SYSTEMS
PERFORMING CATEGORY A FUNCTIONS**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 60880 has been prepared by subcommittee 45A: Instrumentation, control and electrical power systems of nuclear facilities, of IEC technical committee 45: Nuclear instrumentation. It is an International Standard.

This third edition cancels and replaces the second edition published in 2006. It constitutes a technical revision.

The revision of the standard is intended to accomplish the following:

- To take into account the fact that software engineering techniques advanced significantly in the intervening years.
- To align the standard with the new IAEA document SSG-39.
- To align as far as possible the definitions in the standard with the IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition.
- To adapt the requirements associated with standards published or revised since the second edition of IEC 60880, especially IEC 61513, IEC 61226, IEC 62138, IEC 60987, IEC 62340, IEC 62645 and IEC 62671.

- To review the existing requirements.

The text of this standard is based on the following documents:

FDIS	Report on voting
45A/XXX/FDIS	45A/XXX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

INTRODUCTION

a) Technical background and main issues

Engineering of software based Instrumentation and Control (I&C) systems to be used for nuclear safety purposes is a challenge due to the safety requirements to be fulfilled. The safety software used in nuclear power plants (NPP) which are often required only in emergency cases, have to be fully validated and qualified before being used in operation. In order to achieve the high reliability required, special care has to be taken throughout the entire life cycle, from the basic requirements, the various design phases and V&V procedures for operation and maintenance. It is the main aim of this standard to address the related safety aspects and to provide requirements for achieving the high software quality necessary.

It is intended that the standard be used by systems developers, systems purchasers/users (utilities), equipment family providers, systems assessors and by licensors.

b) Situation of the current standard in the structure of the SC 45A standard series

IEC 60880 is directly referenced by IEC 61513 which deals with the system aspects of instrumentation and control systems and equipment that are used to perform functions important to safety in nuclear power plants (NPPs).

IEC 60880 is the second level SC 45A document tackling the issue of software aspects for I&C systems performing category A functions.

Software for categories B and C functions is dealt with in IEC 62138.

IEC 60880 and IEC 62138 together cover the domain of the software aspects of computer-based systems used in nuclear power plants to perform functions important to safety.

IEC 62566 and IEC 62566-2 together cover the domain of the programmable logic used in nuclear power plants to perform functions important to safety.

IEC 62340 covers the domain of requirements for coping with common cause failure (CCF).

This third edition of IEC 60880 is to be read in conjunction with IEC 60987 and IEC 61226, the appropriate SC 45A standards on hardware and on functions categorization.

For more details on the structure of the SC 45A standard series see item d) of this introduction.

c) Recommendation and limitation regarding the application of this standard

It is important to note that this standard establishes no additional functional requirements for systems important to safety.

Aspects for which special requirements and recommendations have been produced, are:

- 1) a general approach to software development to assure the production of the highly reliable software required including hardware and software interdependencies;
- 2) a general approach to software verification and to the software aspects of the computer-based system validation;
- 3) procedures for software modification and configuration control;
- 4) requirements for use of tools;
- 5) procedures for qualification of pre-developed software.

It is recognised that software technology is continuing to develop at a rapid pace and that it is not possible for a standard such as this to include references to all modern design technologies and techniques.

To ensure that the standard will continue to be relevant in future years the emphasis has been placed on issues of principle, rather than specific software technologies.

If new techniques are developed then it is deemed possible to assess the suitability of such techniques by applying the safety principles contained within this standard.

d) Description of the structure of the SC 45A standard series and relationships with other IEC documents and other bodies documents (IAEA, ISO)

The top-level documents of the IEC SC 45A standard series are IEC 61513 and IEC 63046. IEC 61513 provides general requirements for I&C systems and equipment that are used to perform functions important to safety in NPPs. IEC 63046 provides general requirements for electrical power systems of NPPs; it covers power supply systems including the supply systems of the I&C systems. IEC 61513 and IEC 63046 are to be considered in conjunction and at the same level. IEC 61513 and IEC 63046 structure the IEC SC 45A standard series and shape a complete framework establishing general requirements for instrumentation, control and electrical systems for nuclear power plants.

IEC 61513 and IEC 63046 refer directly to other IEC SC 45A standards for general topics related to categorization of functions and classification of systems, qualification, separation, defence against common cause failure, control room design, electromagnetic compatibility, cyber security, software and hardware aspects for programmable digital systems, coordination of safety and security requirements and management of ageing. The standards referenced directly at this second level are to be considered together with IEC 61513 and IEC 63046 as a consistent document set.

At a third level, IEC SC 45A standards not directly referenced by IEC 61513 or by IEC 63046 are standards related to specific equipment, technical methods, or specific activities. Usually these documents, which make reference to second-level documents for general topics, can be used on their own.

A fourth level extending the IEC SC 45 standard series, corresponds to the Technical Reports which are not normative.

The IEC SC 45A standards series consistently implements and details the safety and security principles and basic aspects provided in the relevant IAEA safety standards and in the relevant documents of the IAEA nuclear security series (NSS). In particular this includes the IAEA requirements SSR-2/1, establishing safety requirements related to the design of nuclear power plants (NPPs), the IAEA safety guide SSG-30 dealing with the safety classification of structures, systems and components in NPPs, the IAEA safety guide SSG-39 dealing with the design of instrumentation and control systems for NPPs, the IAEA safety guide SSG-34 dealing with the design of electrical power systems for NPPs and the implementing guide NSS17 for computer security at nuclear facilities. The safety and security terminology and definitions used by SC 45A standards are consistent with those used by the IAEA.

IEC 61513 and IEC 63046 have adopted a presentation format similar to the basic safety publication IEC 61508 with an overall life-cycle framework and a system life-cycle framework. Regarding nuclear safety, IEC 61513 and IEC 63046 provide the interpretation of the general requirements of IEC 61508-1, IEC 61508-2 and IEC 61508-4, for the nuclear application sector. In this framework IEC 60880 and IEC 62138 correspond to IEC 61508-3 for the nuclear application sector. IEC 61513 and IEC 63046 refer to ISO as well as to IAEA GS-R part 2 and IAEA GS-G-3.1, and IAEA GS-G-3.5 for topics related to quality assurance (QA). At level 2, regarding nuclear security, IEC 62645 is the entry document for the IEC/SC 45A security standards. It builds upon the valid high level principles and main concepts of the generic security standards, in particular ISO/IEC 27001 and ISO/IEC 27002; it adapts them and

89 completes them to fit the nuclear context and coordinates with the IEC 62443 series. At level
90 2, IEC 60964 is the entry document for the IEC/SC 45A control rooms standards and IEC 62342
91 is the entry document for the ageing management standards.

92 NOTE It is assumed that for the design of I&C systems in NPPs that implement conventional safety functions (e.g.
93 to address worker safety, asset protection, chemical hazards, process energy hazards) international or national
94 standards would be applied.

95

NUCLEAR POWER PLANTS – INSTRUMENTATION AND CONTROL SYSTEMS IMPORTANT TO SAFETY – SOFTWARE ASPECTS FOR COMPUTER-BASED SYSTEMS PERFORMING CATEGORY A FUNCTIONS

1 Scope

This International Standard provides requirements for the software of computer-based I&C systems of nuclear power plants performing functions of safety category A as defined by IEC 61226.

According to the definition in IEC 61513, I&C systems of safety class 1 are basically intended to support category A functions, but may also support functions of lower categories. However the system requirements are always determined by the functions of the highest category implemented.

For software of I&C system performing only category B and C functions in NPP as defined by IEC 61226, the requirements and guidance of IEC 62138 are applicable.

This standard provides requirements for the purpose of achieving highly reliable software. It addresses each stage of software generation and documentation, including requirements specification, design, implementation, verification and validation.

The principles applied in developing these requirements include:

- best available practices;
- top-down design methods;
- modularity;
- verification of each phase;
- clear documentation;
- auditable documents;
- validation testing.

Annex A provides detailed requirements and recommendations for design and implementation of software.

Additional informative guidance is given in Annex B to Annex G.

This document does not explicitly address how to protect software against those threats arising from malicious attacks, i.e. cybersecurity, for computer-based systems. IEC 62645 provides cybersecurity requirements.

IEC 62671 may be used as an alternative to this document for digital devices of limited functionality. IEC 62671 contains precise criteria to determine if it is applicable to a particular device.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- 137 IEC 60671, *Periodic tests and monitoring of the protection system of nuclear reactors*
- 138 IEC 61226, *Nuclear power plants – Instrumentation, control and electrical power systems*
139 *important to safety – Categorization of functions and classification of systems*
- 140 IEC 61513:202X, *Nuclear power plants – Instrumentation and control important to safety –*
141 *General requirements for systems*
- 142 IEC 62671, *Nuclear power plants – Instrumentation and control important to safety – Selection*
143 *and use of industrial digital devices of limited functionality*
- 144 IEC 62645, *Nuclear power plants – Instrumentation, control and electrical power systems –*
145 *Cybersecurity requirements*
- 146 IAEA SSG-39, *Design of Instrumentation and Control Systems for Nuclear Power Plants*

147 3 Terms and definitions

148 For the purposes of this document, the following terms and definitions apply.

149 3.1

150 animation

151 process by which the behaviour defined by a specification is displayed with actual values
152 derived from the stated behaviour expressions and from some input values

153 3.2

154 application function

155 function of an I&C system that performs a task related to the process being controlled rather
156 than to the functioning of the system itself

157 [SOURCE: IEC 61513:202X, 3.1]

158 3.3

159 application-oriented language

160 computer language specifically designed to address a certain type of application and to be used
161 by persons who are specialists of this type of application

162 [SOURCE: IEC 62138:2018, 3.4]

163 NOTE 1 Equipment families usually feature application-oriented languages so as to provide easy to use capability
164 for adjusting the equipment to specific requirements.

165 NOTE 2 Application-oriented languages may be used to specify the functional requirements of an I&C system,
166 and/or to specify or design application software. They may be based on texts, on graphics, or on both.

167 NOTE 3 Examples: Instruction List (IL), Structured Text (ST), Ladder Diagram (LD), Function Block Diagram (FBD).

168 NOTE 4 See also general-purpose language.

169 3.4

170 application software

171 software complementing the system software of a system, as needed to implement its applica-
172 tion and service functions

173 [SOURCE: IEC 61513:202X, 3.2]

174 NOTE 1 The distinction between system software (generic, re-usable) and application software (typically plant-
175 specific) is generally made to minimize duplication of code and to optimize the allocation of the qualification activities.
176 It is an especially useful distinction when using an I&C platform.

177 NOTE 2 See also system software, application functions, service function.

178 **3.5**

179 **automated code generation**

180 function of automated tools allowing transformation of the application-oriented language into a
181 form suitable for compilation or execution

182 **3.6**

183 **channel**

184 an arrangement of interconnected components within a system that initiates a single output.

185 [SOURCE: IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition]

186 NOTE A channel loses its identity where single output signals are combined with signals from other channels, e.g.,
187 from a monitoring channel, or a safety actuation channel.

188 **3.7**

189 **code compaction**

190 purposeful reduction in memory size required for a program by the elimination of redundant or
191 extraneous instructions

192 **3.8**

193 **common cause failure (CCF)**

194 failures of two or more structures, systems or components due to a single specific event or
195 cause

196 [SOURCE: IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition]

197 NOTE For example consider software dependent on calendar date or similar time counters used in several systems
198 and comprising a design fault leading to its failure on a specific date. The single initiator leads to the simultaneous
199 common cause failure of several components (all components containing the offending software).

200 **3.9**

201 **computer program**

202 set of ordered instructions and data that specify operations in a form suitable for execution by
203 a microprocessor or microcontroller

204 NOTE This includes traditional programs written in general-purpose languages. This also includes programs written
205 in application-oriented languages.

206 **3.10**

207 **computer-based item**

208 item that relies on software instructions running on microprocessors or microcontrollers

209 [SOURCE: IEC 62138:2018, 3.8]

210 NOTE 1 In this term and its definition, the term item can be replaced by the terms: system or equipment or device.

211 NOTE 2 Computer-based item is a kind of programmable digital item.

212 NOTE 3 This term is equivalent to software-based item.

213 NOTE 4 The IEC definition differs from the IAEA definition for “computer-based systems” because it needs to be
214 more specific on the kind of digital technology which is covered.

215 **3.11**

216 **data**

217 representation of information or instructions in a manner suitable for communication,
218 interpretation, or processing by computers

219 [SOURCE: IEC 61513:202X, 3.13]

220 NOTE Data which are required to define parameters and to instantiate application and service functions in the
221 system are called “configuration data”.

3.12**defence in depth**

a hierarchical deployment of different levels of diverse equipment and procedures to prevent the escalation of anticipated operational occurrences and to maintain the effectiveness of physical barriers placed between a radiation source or radioactive material and workers, members of the public or the environment, in operational states and, for some barriers, in accident conditions

[SOURCE: IAEA Nuclear Safety and Security Glossary, 2022 (Interim) Edition]

3.13**deterministic software**

design principle that allows to know, on the one hand the execution path for each situation envisaged for the software and on the other hand the use of the resources for each of these paths

3.14**diversity**

the presence of two or more independent (redundant) systems or components to perform an identified function, where the different systems or components have different attributes so as to reduce the possibility of common cause failure, including common mode failure

[SOURCE: IAEA Nuclear Safety and Security Glossary, 2022 (Interim) Edition]

NOTE When "Diversity" is used with an additional attribute, the term diversity indicates the general meaning "the presence of two or more different ways or means of achieving a specified objective", where the attribute indicates the characteristics of the different ways applied, e.g. functional diversity, equipment diversity, signal diversity.

3.15**dynamic analysis**

process of evaluating a system or component based on its behaviour during execution. In contrast to static analysis

3.16**failure**

loss of the ability of a structure, system or component to function within acceptance criteria

[SOURCE: IAEA Nuclear Safety and Security Glossary, 2022 (Interim) Edition]

NOTE 1 the structure, system or component is considered to fail when it becomes incapable of functioning, whether or not this is needed at that time. A failure in, for example, a backup system may not be manifest until the system is called upon to function, either during testing or on failure of the system it is backing up.

NOTE 2 A failure may be the result of, for example, a hardware fault, a software fault, a system fault, an operator error or a maintenance error.

NOTE 3 See also fault.

3.17**fault**

defect in a hardware, software or system component

[SOURCE: IEC 61513:202X, 3.22]

NOTE 1 Faults may be originated from random defects, that result e.g. from hardware degradation due to ageing, and may be systematic faults, e.g. software faults, which result from design errors.

NOTE 2 A fault (notably a design fault) may remain undetected in a system until specific conditions are such that the result produced does not conform to the intended function, i.e. a failure occurs.

3.18**fault tolerance**

built-in capability of a system to provide continued correct execution in the presence of a limited number of hardware or software faults

3.19**firmware**

software which is closely coupled to the hardware characteristics on which it is installed

[SOURCE: IAEA Nuclear Safety and Security Glossary, 2022 (Interim) Edition]

3.20**general-purpose language**

computer language designed to address all types of usage

[SOURCE: IEC 62138:2018, 3.23]

NOTE 1 The operational system software of equipment families is usually implemented using general-purpose languages.

NOTE 2 Examples: Ada, C.

NOTE 3 See also application-oriented language.

3.21**human error**

discrepancy between the human action taken or omitted, and that intended or required

[SOURCE: IEC 61513:202X, 3.29]

3.22**initialise**

to set counters, switches, addresses, or contents of storage devices to zero or other starting values at the beginning of, or at prescribed points in, the operation of a computer program

3.23**integration tests**

tests performed during the integration of several verified software modules, or during the hardware/software integration process prior to computer-based system validation to verify compatibility of the software and the computer hardware

3.24**library**

collection of related software elements that are grouped together, but which are individually selected for inclusion in the final software product

3.25**operational system software**

part of the system software running on the target processor during system operation

EXAMPLE Operating system, input/output drivers, exception handler, communication software, application software libraries, self-supervision, redundancy and graceful degradation management.

NOTE See also system software.

3.26**pre-developed software (PDS)**

software that already exists, is available as a commercial or proprietary product or as open-source software, and is being considered for use in an instrumentation and control system

[SOURCE: IAEA Nuclear Safety and Security Glossary, 2022 (Interim) Edition, modified]

NOTE The IEC definition differs from the IAEA definition of “predeveloped item” because open-source software have been explicitly added as candidate pre-developed software.

3.27**programmable digital item**

item that relies on software instructions or programmable logic to accomplish a function

315 [SOURCE: IEC 62138:2018, 3.34]

316 NOTE 1 In this term and its definition, the term item can be replaced by the terms: system or equipment or device.

317 NOTE 2 The main kinds of programmable digital items are computer-based items and programmable logic items.

318 NOTE 3 This term used by IEC SC 45A is equivalent to programmable electronic item (PE item) defined according
319 to IEC 61508.

320 **3.28**

321 **programmable logic item**

322 item that relies on logic components with an integrated circuit that consists of logic elements
323 with an inter-connection pattern, parts of which are user programmable

324 [SOURCE: IEC 62138:2018, 3.35]

325 NOTE 1 In this term and its definition, the term item can be replaced by the terms: system or equipment or device.

326 NOTE 2 A programmable logic item is a kind of programmable digital item.

327 **3.29**

328 **redundancy**

329 provision of alternative (identical or diverse) structures, systems and components, so that any
330 single structure, system or component can perform the required function regardless of the state
331 of operation or failure of any other

332 [SOURCE: IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition]

333 **3.30**

334 **safety function**

335 a specific purpose that must be accomplished for safety for a facility or activity to prevent or to
336 mitigate radiological consequences of normal operation, anticipated operational occurrences
337 and accident conditions

338 [SOURCE: IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition]

339 NOTE 1 Safety functions are required to fulfil the fundamental safety functions that are: (i) control of reactivity, (ii)
340 removal of heat from the reactor and from the fuel store, (iii) confinement of radioactive material, shielding against
341 radiation and control of planned radioactive releases, as well as limitation of accidental radioactive releases [IAEA
342 SSR-2/1 2016, Requirement 4].

343 NOTE 2 Safety functions are primarily those that are credited in the safety analysis and include functions performed
344 at all levels of defence in depth.

345 **3.31**

346 **service function**

347 function of an I&C system related to the functioning of the system itself, such as configuration,
348 commissioning, periodic testing, self-supervision, maintenance, modification and security

349 [SOURCE: IEC 61513:20XX, 3.57]

350 NOTE See also application function.

351 **3.32**

352 **software**

353 programs (i.e. sets of ordered instructions), data, rules and any associated documentation
354 pertaining to the operation of a computer-based I&C system

355 [SOURCE: IEC 62138:2018, 3.37]

356 NOTE The term is reserved for microprocessor / microcontroller based technology. Interconnection information and
357 other configuration data representing the implementation of programmable logic devices are not included.

3.33**software development**

all activities of the software lifecycle that leads to the creation of the software of an I&C system or of a software product and that covers all the phases from software requirements specification to validation and installation on site

[SOURCE IEC 62138:2018, 3.39]

3.34**software modification**

change in an already agreed document(s), program or data leading to an alteration of the executable code

NOTE 1 Software modifications may occur either during initial software development (for example to remove faults found in later stages of development), or after the software is already in service.

NOTE 2 During the initial software development, any change leading to an alteration of the executable code that needs to be performed on outputs of a development phase before the phase has been formally completed is not considered software modification. In contrary, any change leading to an alteration of the executable code that needs to be performed on outputs of a development phase after the phase has been formally completed is considered software modification.

3.35**software safety life cycle**

necessary activities involved in the development and operation of the software of an I&C system important to safety occurring during a period of time that starts with the software requirements specification and finishes when the software is withdrawn from use

[SOURCE: IEC 62138:2018, 3.41]

3.36**software version**

instance of a software product derived by modification or correction of a preceding software product instance

3.37**specification**

document that specifies, in a complete, precise, verifiable manner, the requirements, design behaviour or other characteristics of a system or component and, often, the procedures for determining whether these provisions have been satisfied

NOTE There are different types of specifications, for example software requirements specification or software design specification.

3.38**static analysis**

analysis of a system or component based upon its form, structure, content or documentation

[SOURCE: IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition]

NOTE In contrast to dynamic analysis.

3.39**system software**

software designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system and associated programs, for example operational system software and utilities

[SOURCE: IEC 61513: 202X, 3.66]

NOTE 1 Utilities, also called support software comprise software that aids in the development, test, or maintenance of other software and of the system such as translators, graphic editors, diagnostics tools, test and simulation tools.

405 NOTE 2 See also operational system software, application software.

406 **3.40**
407 **system validation**

408 confirmation by examination and provision of evidence that a system fulfils in its entirety the
409 specification of requirements as intended (e.g. validation of an instrumentation and control
410 system in terms of functionality, response time, fault tolerance and robustness)

411 [SOURCE: IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition]

412 **3.41**
413 **verification**

414 confirmation by examination and by means of objective evidence that specified objectives have
415 been met and specified requirements for specific results have been fulfilled

416 [SOURCE: definition 2. IAEA Nuclear Safety and Security Glossary 2022 (Interim) Edition]

417 **4 Symbols and abbreviations**

418 CASE Computer Aided Software Engineering

419 CCF Common Cause Failure (see 3.8)

420 I&C Instrumentation and Control

421 PDS Pre-Developed Software (see 3.26)

422 V&V Verification and Validation

423 **5 General requirements for software projects**

424 **5.1 General**

425 The process of producing instrumentation and control systems important to safety for use in
426 nuclear power plants is set down in IEC 61513 that introduces the concept of the system safety
427 lifecycle as a vehicle by which the development process can be controlled and whose adoption
428 also results in the evidence necessary to justify the operation of systems important to safety.
429 The system safety lifecycle described in IEC 61513 includes and places requirements on, but
430 does not dictate, the project arrangements to be used for production of systems (see Figure 1).

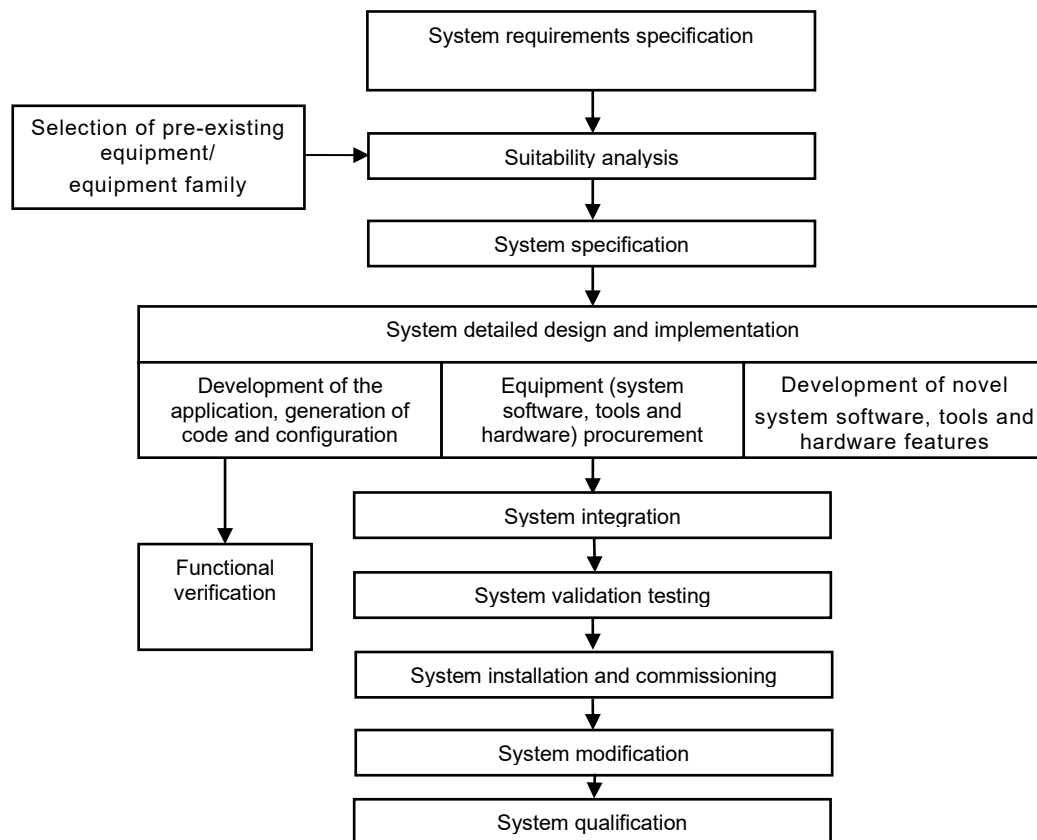


Figure 1 – Activities of the system safety lifecycle (as defined by IEC 61513)

For computer-based systems the system safety lifecycle is further developed to introduce the concept of a software safety lifecycle (see Figure 2 for activities). This shows the hardware and software development being undertaken in parallel from a common specification but coming together at the integration and installation phases of the lifecycle.

The following processes support the phased development process of producing the software:

- software project management (5.4);
- software quality assurance plan (5.5);
- software configuration management (5.6);
- selection of languages (7.2);
- software verification (8);
- defences against common cause failure due to software (13);
- software tools for the development of software (14).

The resulting software related activities in the system safety lifecycle and the supporting processes are shown below in Figure 2 (boxes in bold lines).

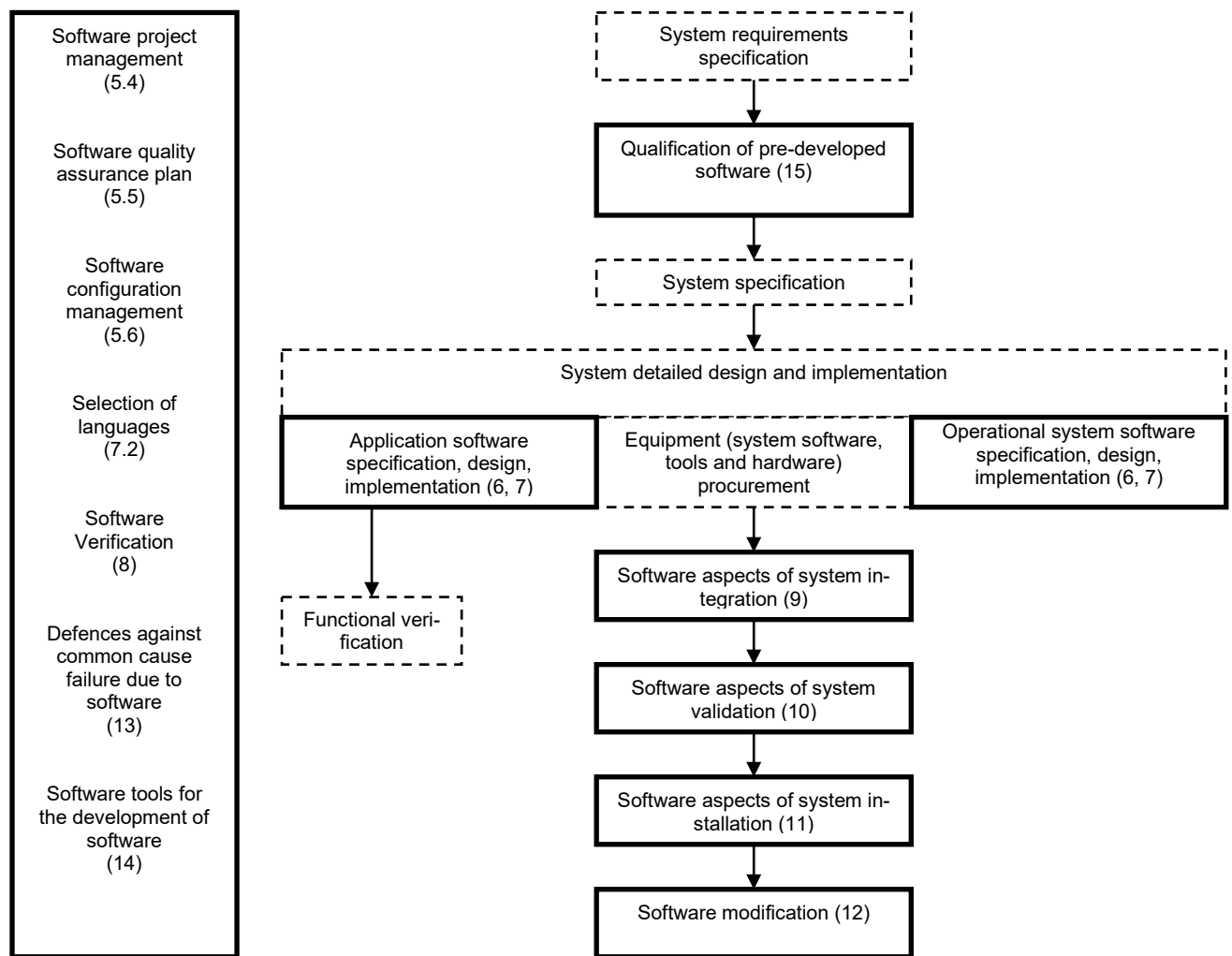


Figure 2 – Software related activities in the system safety lifecycle

(boxes in thin dotted lines represent system activities not addressed in this standard)

Subclauses 5.2 and 5.3 introduce the different software types and the development approach considered in this standard.

5.2 Software types

The software components of a system are often defined as being either operational system software (communications, I/O management, standard functions, self-supervision, etc.) or application software (interlock logic, control loops, display formats, alarm logic, etc.).

Application software generally uses the facilities provided by the operational system software, thus reducing the need for duplication of code within modules and thus reducing the overall amount of software.

Application software is usually specific to one project.

Operational system software may be used in different projects.

Many system designs make extensive use of configuration data. Configuration data may be associated with operational system software or with application software. Configuration data associated with application software consists mainly of plant engineering data resulting from

the design of the plant, and is often prepared by plant designers who are not required to have software skills.

Many modern I&C equipment families are provided with extensive development tools which enable system engineers to design and produce executable codes.

As an example, a typical I&C system developed using components of an equipment family, includes the following:

- pre-developed software components, such as the operational system software kernel and the application function libraries. Generally, these components have been developed using general-purpose languages;
- configuration data needed to adapt the operational system software kernel to the I/O environment and to the services required by the application;
- application software developed using an application-oriented language;
- configuration data needed to parameterize the application software. A subset of the configuration data may have to be modified on site during plant operation (for example, alarm limits, set points, data required to calibrate instrumentation).

5.3 Software development approach

Software usually contributes strongly to the functions performed by the I&C system. Software may also support additional service functions introduced by system design (e.g. configuration, commissioning, periodic testing, self-supervision, maintenance, modification and security). Thus, the software safety lifecycle is in most cases strongly integrated with the system safety lifecycle. In particular, the software requirements specification is a part of, or is derived directly from, system requirements specification and system design.

Though the verification of software components is definitely a part of the software safety lifecycle, there is often no separate and well-identified boundary between software integration and system integration. Therefore, in this standard, software integration is considered to be a part of system integration. Software validation too is not a purely software activity: in this standard, it is considered a part of system integration and/or system validation.

The standard assumes that the software life cycle, originally intended for the development of software with general-purpose languages, is also applicable to application-oriented languages, configuration of pre-developed software and management of configuration data.

However, the standard recognises differences in the development process by introducing dedicated sub-processes for each software type at the implementation level:

- implementation using general-purpose languages;
- implementation using application-oriented languages with associated code generators;
- selection, use and configuration of pre-developed software products;
- management of configuration data.

As boxes “Application software specification, design, implementation” and “Operational system software specification, design, implementation” in Figure 2 represent a large and essential part of the software safety lifecycle, a “zoom” is provided in Figure 3, which illustrates in more detail the activities between software requirements definition and software aspects of system validation, with a clear representation of the four different implementation paths. In that figure, references to related subclauses of this standard are in brackets.

Annex A provides detailed requirements and recommendations for design and implementation of software.

The principles, reflected in the requirements of this standard, are related to the quality of the final code, and are applicable regardless of whether the code is developed using general-

510 purpose languages, application-oriented languages with automated code generation,
 511 configuration of pre-developed software or configuration data.

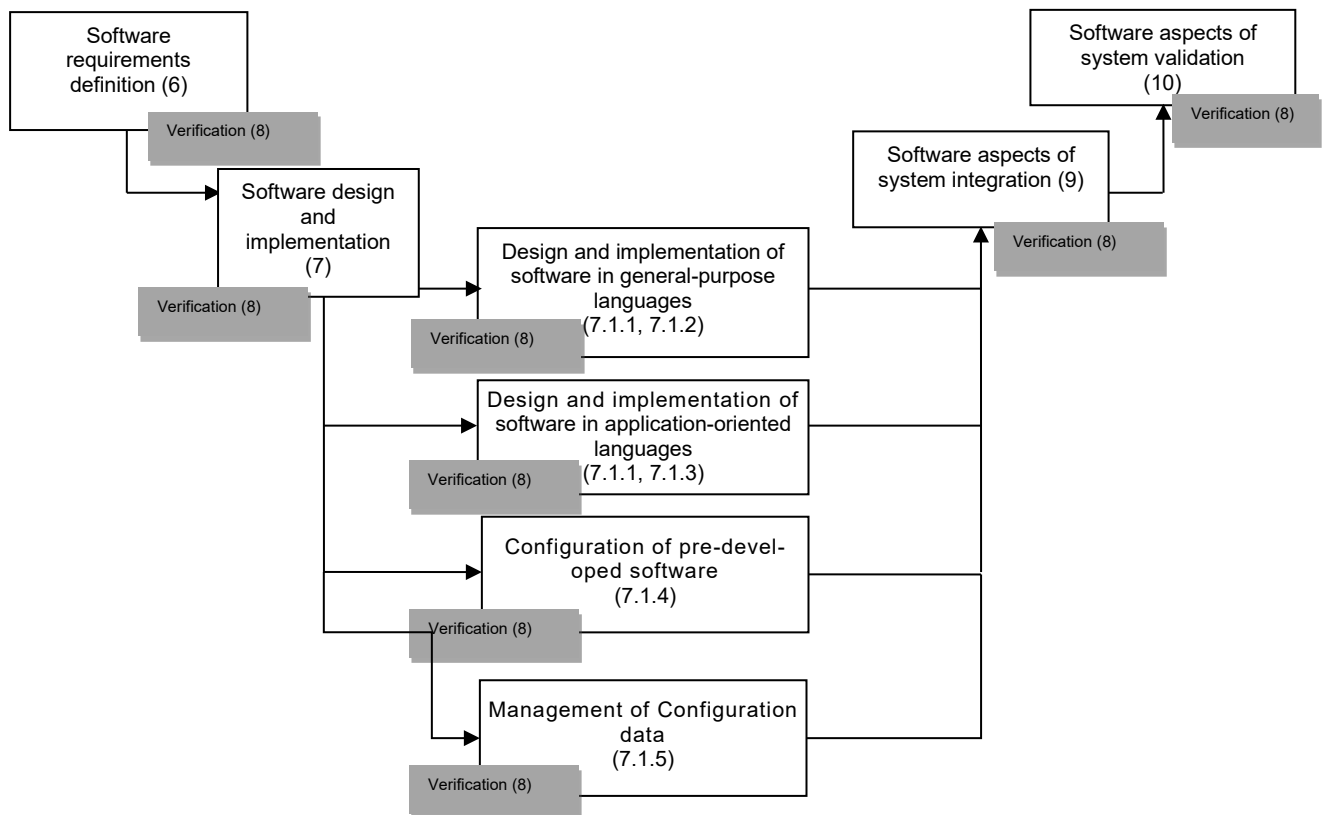


Figure 3 – Development activities of the IEC 60880 software safety lifecycle

5.4 Software project management

5.4.1 Any software project shall be structured into a number of phases.

Each phase is to some extent self-contained but will depend on other phases and will, in its turn, be depended on by others. These phases are informally recognizable by the specific activities pertinent to them.

The phases and associated activities defined for a software project describe a process called in this standard the "software development".

5.4.2 The approach to software development shall be based on the traditional "V" model.

This approach has been reflected and promulgated in other standards notably IAEA SSG-39. It is however recognised that some phases of the development can be done automatically by tools and that software development is an iterative process.

5.4.3 A phase may start before the activities of the preceding phase are complete; however, a phase shall only be terminated if the preceding phases have been completed and if its outputs are consistent with the inputs provided by these preceding activities.

The following general factors determine the activities and phases in implementing a software project.

5.4.4 The activities performed during the phases of the development process shall be identified, according to the software development approach chosen for the project (see 5.2 and 5.3).

5.4.5 Software development activities shall address the whole software safety lifecycle.

5.4.6 Each phase of the software development identified in 5.4.1 shall be divided into well-defined activities.

5.4.7 The phases of the software development shall be formalised and none of the identified phases shall be omitted.

5.4.8 When activities of the software development are automated using software tools, the activities automated shall be documented including the documentation of the inputs and outputs relevant to the phase.

5.4.9 The inputs and outputs of each phase shall be defined and documented.

5.4.10 Every output of each phase shall be systematically checked (item c) of A.1 and item g) of A.4).

5.4.11 Each phase shall include generation of the appropriate documents.

5.4.12 Each phase shall be systematically terminated by a review including examination of relevant documents.

5.4.13 A list of documents required through the software safety life cycle shall be established during the software development.

A typical list of documents is given in Annex D and the document titles from this list are used in the definition of several requirements. The list of documents is however informative as users of the standard have the freedom to structure the documentation as they wish provided that the requirements are satisfied. In particular, some software documents may be omitted if their content is fully included in system level documents.

5.5 Software quality assurance plan

5.5.1 A software quality assurance plan shall exist or be established at an early stage of the software safety life cycle.

Special quality assurance plans may be adopted for individual product phases or particular software components according to national or company standards provided the principles defined in this standard are addressed.

5.5.2 All technical procedures required during each phase of the software safety life cycle shall be addressed by the software quality assurance plan.

5.5.3 The software quality assurance plan shall identify the general roles, their responsibilities and their level of independence (if requested) applied for the project.

5.5.4 The software quality assurance plan shall require that the implementation of the activities of the phases is assigned to competent persons equipped with adequate resources.

5.5.5 The software quality assurance plan shall require that modifications in already approved documents are identified, reviewed and approved by authorized persons.

5.5.6 The software quality assurance plan shall require that the methods, languages, tools, rules and standards used are identified and documented, and known to, and mastered by, the persons concerned.

5.5.7 The software quality assurance plan shall require that when several methods, languages, tools, rules and/or standards are used, it is clear which ones were used for each activity.

5.5.8 The software quality assurance plan shall require that project specific terms, expressions, abbreviations and conventions are explicitly defined.

5.5.9 The software quality assurance plan shall require that any quality issues raised are tracked and resolved.

5.5.10 The software quality assurance plan shall require that records resulting from its application are produced.

5.5.11 Every verification step or review shall result in a report on the analysis performed, the conclusions reached and the resolutions agreed. This report shall be included in the documentation.

5.5.12 Any deviation from the software quality assurance plan shall be documented and justified.

5.6 Software configuration management

Subclause 6.3.2.3 (system configuration management plan) of IEC 61513 provides requirements for configuration management at the I&C system level. This subclause provides additional requirements specific, or of particular importance, to software.

5.6.1 Configuration management for software shall be performed according to the provisions of a software configuration management plan.

5.6.2 These provisions shall be consistent with those for system level configuration management.

5.6.3 Software configuration management plan shall be established early in the lifecycle of a software project to address the following requirements.

5.6.4 Each produced version of each software entity shall be uniquely identified.

5.6.5 It shall be possible to identify the relevant versions of all software documentation associated with each software entity.

5.6.6 Software under development shall be segregated from software which has achieved release or verified status.

5.6.7 It shall be possible to identify the versions of all software entities which together constitute a complete version of the final product.

5.6.8 It shall be possible to verify the integrity of the software entities.

5.6.9 It shall be possible to identify the version of the software in the target system.

5.6.10 It shall be possible to retrospectively identify all software entities affected by the implementation of a modification.

5.6.11 It shall be possible to identify all translation tools and tool versions used to produce each executable entity (see 14.3.3).

6 Software requirements definition

6.1 Specification of software requirements

6.1.1 The software requirements shall be derived from requirements at system level.

6.1.2 The software requirements shall describe what the software has to do and not how the software shall do it.

6.1.3 The software requirements shall specify:

- 1) the application functions to be provided by the software;
- 2) the different modes of behaviour of the software, and the corresponding conditions of transition;
- 3) the interfaces and interactions of the software with its environment (e.g. with operators, with the rest of the I&C system, with the other systems, if any, with which it interacts or shares resources), including the roles, types, formats, ranges and constraints of inputs and outputs;
- 4) the interface between functions of different levels of safety significance (e.g. category A and non-category A functions) performed by the software;
- 5) the parameters of the software which can be modified manually during operation, if any, their roles, types, formats, ranges and constraints, and the checks to be performed by the software when they are modified;
- 6) the required software performance, in particular response time requirements;
- 7) what the software must not do or must avoid, when appropriate;
- 8) the requirements of, or the assumptions made by, the software regarding its environment, when applicable;
- 9) the requirements if any, of standard software packages;
- 10) the requirements specifying mitigation of fault propagation.

6.1.4 The process of laying down software requirements shall be documented.

6.1.5 The following constraints between hardware and software shall be described:

- 1) operating characteristics in general (word length, exchange types, speed, etc.);

This may be achieved through a reference to the equipment manufacturer manuals when this one is sufficiently detailed.

- 2) special equipment operating characteristics (particular drivers, data-communications equipment, startup, etc.);
- 3) arithmetic constraints;
- 4) requirements of hardware supervision.

6.1.6 A reference to the hardware requirements specification shall be made within the software requirements for any impacts from hardware design.

6.1.7 Special operating circumstances such as plant commissioning and refuelling shall be described down to the software level for the functions that are impacted.

6.2 Self-supervision

6.2.1 The software shall supervise itself and the hardware during operation within specified time intervals.

This is considered to be a primary factor in achieving high overall system reliability.

6.2.2 Those parts of the memory that contain code or invariable data shall be monitored to detect unintended changes.

6.2.3 The self-supervision should be able to detect to the extent practicable:

- 1) random failure of hardware components;
- 2) erroneous behaviour of software (e.g. deviations from specified software processing and operating conditions or data corruption);
- 3) erroneous data transmission between different processing units.

6.2.4 If a failure is detected by the software during plant operation, the software shall take predefined appropriate and timely response. Those shall be implemented according to the system reactions required by the specification and to IEC 61513 system design rules.

6.2.5 Appropriate automated actions are to be taken at detected failures considering the following factors:

- 1) predefined output shall be guaranteed as far as possible in order to reach failsafe system behavior;
- 2) the consequences of failures shall be minimised;
- 3) remedial routines, such as fall back, system recovery, shall be considered for inclusion;
- 4) sufficiently detailed information on failures shall be provided for the operating staff.

6.2.6 Self-supervision shall not adversely affect the intended system functions and shall not prevent timely system reactions in any circumstance.

6.2.7 It should be possible to automatically collect all useful diagnostic information arising from software self-supervision.

6.3 Periodic testing

For computer-based systems important to safety, the general principles of IEC 60671 apply for those hardware components which are not covered adequately by self-supervision. Hardware components whose failure modes are fully covered by self-supervision does not necessitate periodic testing.

6.3.1 The software shall be designed so as to meet the requirements of periodic testing which takes place within specified maximum intervals (e.g. shut-down periods):

- 1) every safety function shall be coverable by periodic testing;
- 2) any failure of the safety functions shall be detected.

6.3.2 It should be possible to automatically collect all useful diagnostic information arising from software periodic testing.

6.3.3 The quality of the software of auxiliary devices for testing that cannot disturb the class 1 systems shall correspond to the quality of equipment used for verification and validation as per clause 14.

The software dedicated to auxiliary devices for testing of the class 1 systems need not be designed to comply with all software safety requirements of this standard provided that it cannot disturb the class 1 systems during regular operations.

6.3.4 In case non disturbance during regular operations cannot be established, software dedicated to auxiliary devices for testing of the class 1 systems shall be considered class 1 software and the requirements of this standard apply.

6.4 Documentation

6.4.1 The requirements for the software of an I&C system shall be documented.

The corresponding document or set of documents is called the software requirements specification.

6.4.2 The software requirements specification should contain aspects of minor importance to software development which are, however, a background for licensing.

Such important aspects for licensing may be:

- risk considerations;
- recommendations for functions or engineered safety features;
- other items that provide the background for specific requirements;
- special regulatory requirements on software structure, code analysis, V&V, etc.

6.4.3 The software requirements specification shall be presented according to a standard whose formality does not preclude readability for all user groups.

The software requirements specification is intended to be used by:

- requirements preparers;
- the customer, client or final user;
- the software development team;
- the software verification team;
- assessors and licensing personnel.

6.4.4 The software requirements specification shall be unequivocal, testable or verifiable, and achievable.

A formal language or an application-oriented language may be used to improve the coherence and completeness of aspects of the software requirements specification.

Automated tools may be used for this purpose.

6.4.5 The presentation of the software requirements specification shall be detailed sufficiently, free from contradiction, and without duplication as far as possible.

6.4.6 The software requirements specification shall be complete, consistent and up-to-date.