RAPPORT TECHNIQUE

TECHNICAL REPORT

CEI
IEC

61940

Première édition
First edition
1998-06

Instrumentation nucléaire –
Revue de l'application de la CEI 60880 (1986)

Nuclear instrumentation –
A review of the application of IEC 60880 (1986)

Numéro de référence
Reference number
CEI/IEC 61940:1998

Numéros des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60 000.

Publications consolidées

Les versions consolidées de certaines publications de la CEI incorporant des amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles dans le Catalogue de la CEI.

Les renseignements relatifs à ces révisions, à l'établissement des éditions révisées et aux amendements peuvent être obtenus auprès des Comités nationaux de la CEI et dans les documents ci-dessous:

- **«Site web»* de la CEI**
- **Catalogue des publications de la CEI**
  Publié annuellement et mis à jour régulièrement (Catalogue en ligne)*
- **Bulletin de la CEI**
  Disponible sur le «site web»* de la CEI et comme périodique imprimé

Terminologie, symboles graphiques et littéraux

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 60050: *Vocabulaire Electrotechnique International* (VEI).

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera la CEI 60027: *Symboles littéraux à utiliser en électrotechnique*, la CEI 60417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles,* et la CEI 60617: *Symboles graphiques pour schémas.*

* Voir adresse «site web» sur la page de titre.

Numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60 000 series.

Consolidated publications

Consolidated versions of some IEC publications including amendments are available. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available in the IEC catalogue.

Information on the revision work, the issue of revised editions and amendments may be obtained from IEC National Committees and from the following IEC sources:

- **IEC web site***
- **Catalogue of IEC publications**
  Published yearly with regular updates (On-line catalogue)*
- **IEC Bulletin**
  Available both at the IEC web site* and as a printed periodical

Terminology, graphical and letter symbols

For general terminology, readers are referred to IEC 60050: *International Electrotechnical Vocabulary* (IEV).

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications IEC 60027: *Letter symbols to be used in electrical technology*, IEC 60417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets* and IEC 60617: *Graphical symbols for diagrams.*

* See web site address on title page.

RAPPORT TECHNIQUE – TYPE 3

TECHNICAL REPORT – TYPE 3

CEI
IEC

61940

Première édition
First edition
1998-06

Instrumentation nucléaire – Revue de l'application de la CEI 60880 (1986)

Nuclear instrumentation – A review of the application of IEC 60880 (1986)

International Electrotechnical Commission 3, rue de Varembé Geneva, Switzerland
Telefax: +41 22 919 0300 e-mail: inmail@iec.ch IEC web site http: //www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE U

Pour prix, voir catalogue en vigueur
For price, see current catalogue

SOMMAIRE

CONTENTS

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

INSTRUMENTATION NUCLÉAIRE – REVUE DE L'APPLICATION DE LA CEI 60880 (1986)

AVANT-PROPOS

1) La CEI (Commission Electrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI, entre autres activités, publie des Normes internationales. Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de la CEI concernant les questions techniques, représentent, dans la mesure du possible un accord international sur les sujets étudiés, étant donné que les Comités nationaux intéressés sont représentés dans chaque comité d'études.
3) Les documents produits se présentent sous la forme de recommandations internationales. Ils sont publiés comme normes, rapports techniques ou guides et agréés comme tels par les Comités nationaux.
4) Dans le but d'encourager l'unification internationale, les Comités nationaux de la CEI s'engagent à appliquer de façon transparente, dans toute la mesure possible, les Normes internationales de la CEI dans leurs normes nationales et régionales. Toute divergence entre la norme de la CEI et la norme nationale correspondante doit être indiquée en termes clairs dans cette dernière.
5) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand un matériel est déclaré conforme à l'une de ses normes.
6) L'attention est attirée sur le fait que certains des éléments de la présente Norme internationale peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

La tâche principale des comités d'études de la CEI est d'élaborer des Normes internationales. Exceptionnellement, un comité d'études peut proposer la publication d'un rapport technique de l'un des types suivants:

– type 1, lorsque, en dépit de maints efforts, l'accord requis ne peut être réalisé en faveur de la publication d'une Norme internationale;
– type 2, lorsque le sujet en question est encore en cours de développement technique ou lorsque, pour une raison quelconque, la possibilité d'un accord pour la publication d'une Norme internationale peut être envisagée pour l'avenir mais pas dans l'immédiat;
– type 3, lorsqu'un comité d'études a réuni des données de nature différente de celles qui sont normalement publiées comme Normes internationales, cela pouvant comprendre, par exemple, des informations sur l'état de la technique.

Les rapports techniques du type 1 et 2 font l'objet d'un nouvel examen trois ans au plus tard après leur publication afin de décider éventuellement de leur transformation en Normes internationales. Les rapports techniques du type 3 ne doivent pas nécessairement être révisés avant que les données qu'ils contiennent ne soient plus jugées valables ou utiles.

La CEI 61940, rapport technique de type 3, a été établie par le sous-comité 45A: Instrumentation des réacteurs, du Comité d'études 45 de la CEI: Instrumentation nucléaire.

Le texte de ce rapport technique est issu des documents suivants:

Projet de comité	Rapport de vote
45A/244/CDV	45A/284/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de ce rapport technique.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

NUCLEAR INSTRUMENTATION – A REVIEW OF THE APPLICATION OF IEC 60880 (1986)

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the 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, the IEC publishes International Standards. 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. The 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 the 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 National Committees.
3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical report of one of the following types:

– type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
– type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
– type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard, for example “state of the art”.

Technical reports of types 1 and 2 are subject to review within three years of publication to decide whether they can be transformed into International Standards. Technical reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

IEC 61940, which is a technical report of type 3, has been prepared by subcommittee 45A: Reactor instrumentation, of IEC technical committee 45: Nuclear instrumentation.

The text of this technical report is based on the following documents:

Committee draft	Report on voting
45A/244/CDV	45A/284/RVC

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

INTRODUCTION

La CEI 60880 «Logiciel pour les calculateurs utilisés dans les systèmes de sûreté des centrales nucléaires» a été publiée en 1986. La CEI 60880 (souvent citée comme la «Norme» dans ce rapport technique) a été largement utilisée par les développeurs de systèmes pour guider le processus de développement, et par les évaluateurs de systèmes, qui ont utilisé la Norme pour évaluer l'adéquation des systèmes.

Ce rapport fournit une interprétation de la CEI 60880, basée sur le retour d'expérience des développeurs et des évaluateurs, qui nous l'espérons, sera utile aux futurs utilisateurs de la Norme. De plus, si la CEI 60880 (1986) est révisée ultérieurement, on espère que le retour d'expérience décrit dans ce rapport sera pris en compte.

INTRODUCTION

IEC 60880 “Software for computers in the safety systems of nuclear power stations” was published in 1986. IEC 60880 (often referred to as the “Standard” in this report) has been widely used by system developers, to guide the development process, and by system assessors, who have used the standard to assess the fitness for purpose of systems.

This report provides an interpretation of IEC 60880, based upon feedback received from developers and assessors, which it is hoped will benefit future users of the Standard. Also, if IEC 60880 (1986) is revised at some later date, it is expected that the feedback of experience outlined in this report will be taken into account.

INSTRUMENTATION NUCLÉAIRE – REVUE DE L'APPLICATION DE LA CEI 60880 (1986)

1 Domaine d'application et objet

Le présent rapport a pour objet de mettre à profit l'expérience pratique des utilisateurs de la CEI 60880, afin de fournir une interprétation de la CEI 60880 à l'usage des développeurs et des évaluateurs. De nombreuses nations ont apporté leur concours à l'élaboration du présent rapport, dont le contenu est maintenant considéré comme le reflet de l'expérience internationale en ce qui concerne la CEI 60880.

Il n'est pas dans l'intention de ce rapport de représenter une interprétation de la CEI 60880 comme appliquée sur un quelconque projet spécifique. Il est écrit d'une manière générale et il est recommandé de ne pas considérer l'interprétation proposée des articles de la CEI 60880 comme définitive. Ce rapport n'ajoute pas, ne remplace pas et ne modifie pas les exigences de la CEI 60880.

2 Documents de référence

CEI 60880:1986, *Logiciel pour les calculateurs utilisés dans les systèmes de sûreté des centrales nucléaires*

CEI 61508 (toutes les parties), *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité* [1)]

ISO 9001:1994, *Systèmes qualité – Modèle pour l'assurance de la qualité en conception, développement, production, installation et prestations associées*

AIEA 50-SG-D3:1981, *Guide de sûreté – Système de protection et dispositifs associés dans les centrales nucléaires*

3 Définitions et abréviations

Les définitions utilisées dans le présent rapport technique sont données dans l'article 2 «Termes et définitions» de la CEI 60880.

VDU: unité de visualisation

4 Evaluation de la conformité à la CEI 60880

4.1 But de l'évaluation

Une évaluation par rapport à la Norme peut être requise dans de nombreuses situations, par exemple:

a) pendant un processus de développement système, les ingénieurs de conception peuvent évaluer les méthodes et techniques disponibles par rapport à la Norme lorsqu'il s'agit de sélectionner les plus appropriées;

---

1) A publier.

NUCLEAR INSTRUMENTATION – A REVIEW OF THE APPLICATION OF IEC 60880 (1986)

1 Scope and object

The purpose of this report is to combine the practical experience gained by users of IEC 60880 so as to provide an interpretation of IEC 60880 which will be of use to system developers and assessors. During the development of this report contributions have been received from many nations and the contents of this report is now believed to reflect the international experience regarding IEC 60880.

This report is not intended to represent an interpretation of IEC 60880 as applied on any specific project. It is written in a general manner and the suggested interpretations of IEC 60880 clauses should not be taken as definitive. This report does not replace, alter or add to the requirements contained in IEC 60880.

2 Reference documents

IEC 60880:1986, *Software for computers in the safety systems of nuclear power stations*

IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems* [1)]

ISO 9001:1994, *Quality systems – Model for quality assurance in design, development, production, installation and servicing*

IAEA 50-SG-D3:1980, *Safety guide – Protection system and related features in nuclear power plants*

3 Definitions and abbreviations

The definitions used in this technical report are given in clause 2 “Terms and definitions” of IEC 60880.

VDU: visual display unit

4 Assessment of compliance with IEC 60880

4.1 Purpose of assessment

There are many situations where an assessment against the Standard may be required, for example:

a) during a system development process design engineers may assess the available methodologies and technologies against the Standard when deciding which would be the most appropriate to use;

---

1) To be published.

b) pendant la réévaluation de l'argumentaire de sûreté d'une installation existante, la Norme peut être utilisée pour évaluer l'adéquation des systèmes opérationnels;

c) des évaluateurs indépendants peuvent devoir confirmer que le processus de développement est conforme à la Norme.

Le principal objectif de l'évaluation d'un système, ou d'un de ses composants, est de déterminer s'il est manifestement apte à l'emploi. Si l'on peut montrer qu'un système est essentiellement conforme à la CEI 60880, on peut considérer avec certitude qu'il convient pour une utilisation dans un rôle de sûreté nucléaire.

4.2 Processus d'évaluation

Le fait qu'un système soit ou ne soit pas considéré comme «conforme» dépend non seulement de ses caractéristiques et de son cycle de développement, mais aussi de l'aptitude de l'évaluateur à interpréter les exigences et les recommandations fournies par la Norme.

La Norme fournit quelques recommandations pour aider à l'interprétation de son contenu, particulièrement au niveau de l'introduction qui indique que la Norme doit fournir «une approche générale» du développement, de la vérification et de la validation de logiciel; la Norme «traite des principes et prescriptions relatives aux logiciels du système...» et fournit «des recommandations spéciales» pour les systèmes de sûreté. L'utilisation d'une telle terminologie implique que le contenu de la Norme n'est pas fait pour être appliqué de façon dogmatique et qu'il convient de l'interpréter en fonction des exigences pour les applications particulières.

L'article 1, domaine d'application et objet, indique également que les annexes contiennent «des directives et des informations complémentaires sur la manière de satisfaire aux prescriptions de la partie principale de cette norme» – l'accent est mis sur le contenu des articles 3 à 10 plutôt que sur celui des annexes A à F. La Norme exige, en cas d'utilisation de techniques autres que celles identifiées dans les annexes, que celles-ci soient documentées et auditables.

Les articles du texte principal de la Norme sont qualifiés de nombreuses façons, par exemple par «peut», «doit», «il est nécessaire», «il convient de», «serait préférable», «est recommandé». L'importance de toute déviation par rapport à la Norme dépend du degré de contrainte indiqué dans le texte.

A l'issue d'un processus formel d'évaluation, il convient d'obtenir un rapport identifiant les articles particuliers qui sont, de l'avis de l'évaluateur, traités ou non par un système. Il convient de fournir des justifications de conformité documentées, sous forme d'explications, de références à des manuels, de rapports, etc., pour chaque article avec lequel la conformité est affirmée. Quand les exigences/recommandations d'un article n'ont pas été suivies, c'est-à-dire en cas de non-conformité à un article, les classifications de réponse suivantes sont suggérées:

a) non applicable, c'est-à-dire une recommandation de codage non applicable au langage utilisé, ou une exigence dépassant le cadre de l'évaluation, par exemple une exigence applicable au client/aux utilisateurs d'un système au cours de l'évaluation d'un fournisseur de système, ou une exigence système quand le cadre d'une évaluation est limité à l'évaluation du processus de production du logiciel;

b) pas encore applicable, par exemple si une évaluation a été faite avant la livraison d'un système, toutes les recommandations sur l'exploitation du système (article 10 de la CEI 60880) s'inscriraient dans cette catégorie;

c) non souhaitable, c'est-à-dire une recommandation qui, pour une application ou un processus de développement particulier n'améliorerait pas la sûreté/intégrité;

d) non obligatoire, par exemple des recommandations qualifiées avec un «peut»;

b) during the re-assessment of the safety case for an existing station the Standard may be used to assess the adequacy of operational systems;

c) independent assessors may be required to confirm that a development process is compliant with the Standard.

The main purpose of assessing a system, or a system component, is to determine if it is demonstrably fit for use. If a system can be shown to essentially comply with IEC 60880 this provides some considerable confidence that it is suitable for use in a nuclear safety role.

4.2 Assessment process

Whether or not a particular system is considered to be “compliant” depends not only upon the characteristics of the system and its associated development life cycle, but also upon the skill of the assessor in interpreting the requirements and guidance provided by the Standard.

The Standard provides some guidance to assist in the interpretation of its contents, particularly the introduction which states that the Standard is to provide a “general approach” to software development, verification and validation; the Standard “discusses the software principles...” and provides “special recommendations” for safety systems. The use of such terminology implies that the contents of the Standard are not to be applied dogmatically, but that they should be used as guidance and should be interpreted as required for particular applications.

Clause 1, scope and object, also states that the appendices contain “Additional guidance and information on how to comply with the requirements of the main part of this Standard” – the emphasis is placed upon the contents of clause 3 to 10 rather than on those of appendices A to F. The Standard requires that if techniques other than those identified in the appendices are used they are to be documented and auditable.

Clauses in the main body of the Standard are qualified in a number of ways, for example “may”, “shall”, “required”, “should be”, “should be preferred”, “is recommended”. The significance of any deviations from the Standard depends upon the degree of compulsion implied by the text.

The end result of a formal assessment process should be a report identifying the particular clauses which are, in the opinion of the assessor, addressed or not addressed by a system. Documented justifications of compliance, in the form of explanations, references to manuals, reports, etc., should be provided for clauses for which compliance is claimed. Where the requirements/recommendations of a clause have not been followed, i.e. have not been complied with, the following classifications of response are suggested:

a) not applicable, i.e. a coding recommendation not applicable to the language used, or a requirement outside the scope of the assessment, for example a requirement relevant to the purchaser/users of a system when performing an assessment of a system supplier, or a system requirement when the scope of an assessment is limited to assessment of the software production process;

b) not yet applicable, for example if an assessment was performed before a system was delivered, all of the recommendations against system operation (clause 10 of IEC 60880) would fall into this category;

c) not desirable, i.e. a recommendation which, for a particular application or development process, does not enhance safety/integrity;

d) not compulsory, for example recommendations qualified with a “may”;

e) non-conformité mineure, c'est-à-dire un manquement net à la Norme mais qui a une incidence nulle ou mineure sur la qualité/sûreté d'un système par exemple une situation où une autre technique, équivalente (ou meilleure) à celle recommandée dans la Norme, a été utilisée;

f) non-conformité majeure, c'est-à-dire un manquement net à la Norme qui peut avoir des implications significatives sur la qualité/sûreté.

Quand des non-conformités sont identifiées, il convient de fournir une justification incluant une description des facteurs atténuants.

On suggère, pour plus de clarté, d'inclure dans le rapport d'évaluation un chapitre de résumé/conclusions considérant la conformité dans son ensemble, et incluant l'identification des aspects du système «meilleurs» que ceux requis par la CEI 60880. Il convient d'apporter un jugement sur l'adéquation du système à son usage prévu et sur la conformité «générale» avec la Norme. Du fait du large domaine d'application de la Norme, et de la nécessité d'interpréter la Norme pour une application particulière, la détermination de la conformité est rarement binaire et un jugement qualifié peut être nécessaire.

5 Sujets clés de l'évaluation

Il faut prendre en compte un certain nombre de sujets clés lors de l'évaluation de la conformité d'un système.

5.1 Cycle de vie

La Norme définit une progression chronologique, depuis les exigences de l'utilisateur jusqu'à l'obtention du système final, intégré et testé. Chaque phase doit s'achever par une revue critique (voir commentaires sur la CEI 60880, paragraphe 3.1.6 à l'article 6 du présent rapport). En réalité, et particulièrement pour les grands systèmes avec une architecture distribuée, il existe un travail parallèle considérable; le processus de conception tend à être itératif par nature. Les utilisateurs vont identifier de nouvelles exigences pendant le processus de développement, qui doivent être intégrées dans la conception. Il convient qu'aucune non-conformité majeure ne soit levée contre le «timing» des diverses activités de conception, sous réserve que les principes fondamentaux de la Norme soient pris en compte, à savoir que le système final est une manifestation des exigences des utilisateurs et que les processus de vérification/validation sont détaillés, complets, maîtrisés et auditables.

Beaucoup d'exigences de la Norme supposent implicitement un cycle de développement suivant une approche descendante traditionnelle applicable à un nouveau développement. La plupart des «nouveaux» systèmes incluent des composants appartenant à d'anciens systèmes, et pour incorporer de tels composants, un certain degré de conception ascendante est nécessaire.

Un exemple de processus de conception typique pour un système de protection moderne est donné ci-dessous:

a) les utilisateurs développent un ensemble d'exigences fonctionnelles;

b) le fournisseur développe une spécification de conception système décrivant la manière dont les composants standard en provenance d'un «kit» système générique seront utilisés pour fournir la fonctionnalité demandée. Le système générique peut avoir son propre ensemble de documentation de conception et il convient que son développement ait suivi une méthodologie généralement conforme aux recommandations de la CEI 60880 (y compris la vérification indépendante des modules de logiciel, etc.);

c) les fonctionnalités qui demandent un développement de logiciel système sont identifiées et l'ensemble du nouveau logiciel système est développé selon les mêmes normes (ou meilleures) que le logiciel générique existant, voir b) ci-dessus;

e) minor non-compliance, i.e. a clear breach of the Standard but which has nil or minor impact upon the quality/safety of a system for example where an alternative and equivalent (or better) technique has been used to that recommended by the Standard;

f) major non-compliance, i.e. a clear breach of the Standard which may have significant quality/safety implications.

Where non-compliance is identified a justification should be provided, to include a description of any mitigating factors.

It is suggested that, to improve clarity, the assessment report should contain a summary/ conclusions section which considers the extent of overall compliance, including the identification of any aspects of the system which are “better” than that required by IEC 60880. A judgement should be given as to whether a system is fit for its proposed application and if the system is “generally” compliant with the Standard. Due to the broad scope of the Standard, and the need to interpret the Standard for an individual application, the determination of compliance is rarely a “black and white” issue and a qualified judgement may be required.

5 Key assessment issues

There are a number of key issues to be considered when assessing the compliance of a system.

5.1 Life cycle

The Standard defines a chronological progression from user requirements through to the final, integrated and tested system. Each phase is to be completed by a critical review (see comments on IEC 60880, subclause 3.1.6 in clause 6 of this report). In reality, and especially for large systems with a distributed processing architecture, there is usually a considerable degree of parallel working; the design process tends to be iterative in nature. Users will identify new requirements during the development process which have to be included in the design. No major non-compliance should be raised against the timing of various design activities providing that the fundamental principles of the Standard are addressed, namely that the final system is a manifestation of the users' requirements and that the verification/validation processes are comprehensive, controlled and auditable.

Many of the requirements in the Standard implicitly assume a conventional top-down development life cycle for a bespoke system. Most “new” systems include components of old systems and to incorporate such components requires a degree of bottom-up design.

An example of a typical design process for a modern protection system is given below:

a) the users develop a set of functional requirements;

b) the supplier develops a system design specification describing how standard components from a generic system “kit” are to be used to provide the required functionality. The generic system may have its own set of design documentation and it should have been developed under a methodology generally compliant with the recommendations of IEC 60880 (including the independent verification of software modules, etc.);

c) functionality which requires system software development is identified and all new system software is developed to the same (or better) standards as the existing generic software, see b) above;

d) les exigences fonctionnelles du système sont représentées sous forme de diagrammes et transformées en logiciel application en utilisant un outil logiciel validé (ainsi, cet aspect des exigences du système ne serait pas compris dans la documentation de conception classique, basée sur du texte, incluant les exigences système, spécification système, spécification logiciel, etc.);

e) les tests sont réalisés au niveau module logiciel, composant matériel, armoire et système. Les tests au niveau système sont effectués sur un prototype système générique, le système assemblé en usine et de nouveau quand le système est installé sur site (en général deux fois – non connecté puis connecté à la centrale).

Un tel processus de conception n'est pas directement superposable au cycle de développement de la CEI 60880 et il est nécessaire d'interpréter certains articles de la Norme avec prudence pour les appliquer aux parties appropriées du processus de développement.

S'il y a des déviations majeures par rapport au cycle de développement de référence de la CEI 60880, alors la documentation de conception produite sera très différente de ce qui est spécifié dans l'annexe F de la Norme. La Norme est très directive dans ce domaine, mais, sous réserve que les informations demandées par la Norme soient documentées dans les documents de conception du système, aucune non-conformité majeure ne devrait être prononcée.

Si une conception générique, avec un grand nombre de composants système existants, doit être utilisée comme base du nouveau système, tout nouveau composant nécessaire peut être produit en utilisant le langage et les outils de développement initiaux. Toutes les caractéristiques peuvent ne pas être identifiées au paragraphe 5.2 de la Norme. Les avantages inhérents à l'utilisation d'une conception système éprouvée doivent être comparés aux inconvénients de l'utilisation de techniques de développement moins avancées. Sous réserve que l'utilisation des techniques éprouvées soit justifiée de façon adéquate, seule une non-conformité mineure devrait être prononcée (on notera que l'utilisation de logiciel préexistant doit être traitée dans le cadre de nouveaux développements de la CEI 60880).

De nombreux articles de la Norme demandent une interprétation prudente quand on considère des systèmes distribués modulaires contenant un grand nombre de sous-systèmes (composants), chacun d'entre eux contenant un processeur. Il convient d'interpréter dans certains cas le terme système tel qu'il est utilisé dans la Norme comme s'appliquant à l'ensemble du système distribué et dans d'autres cas comme s'appliquant à chacun des sous-systèmes.

Par exemple:

L'intégration matériel/logiciel peut s'effectuer initialement sur une base composant par composant (un composant étant typiquement une carte processeur du type de celles gérant les entrées/sorties d'un système de commande). Un système comprend typiquement de nombreux sous-systèmes comprenant eux-mêmes beaucoup de composants identiques. Il est possible qu'une modification système n'affecte que le fonctionnement interne d'un composant et ainsi, sous réserve que l'interface entre les sous-systèmes n'ait pas changé, une analyse des incidences n'identifierait aucune nécessité de réaliser des essais au-delà de l'essai de contrôle de composant et des tests d'intégration du sous-système. Ainsi, dans ce cas, le sous-système serait considéré en tant que «système» dans la CEI 60880, 7.5.

Il convient de noter qu'un processus d'évaluation peut être extrêmement rallongé si le logiciel développé pour différents sous-systèmes d'un système est développé suivant des cycles de vie différents; dans de telles circonstances, il peut être conseillé de résumer la conformité de différents cycles de vie plutôt que de produire une analyse détaillée de chaque cycle de développement par rapport à chaque article de la Norme.

Ainsi, en résumé, bien que de nombreuses non-conformités puissent être prononcées du fait des différences existant entre le cycle de vie recommandé et le cycle de vie utilisé (et entre le jeu de documents prescrit dans la Norme et le jeu effectivement produit) dans la plupart des cas, les principes que la Norme s'efforce d'imposer sont clairs. Il convient de prononcer des non-conformités majeures uniquement dans le cas de déviations par rapport à ces principes.

d) the system functional requirements are represented diagrammatically and converted into application software using a validated software tool (so this aspect of the system requirements would not be contained within the text-based suite of conventional design documentation of system requirements, system specification, software specification, etc.);

e) testing is performed at software module, hardware component, cubicle and system level. System level testing is performed on a generic system prototype, the factory assembled system and again when the system is installed at site (usually twice – once with the plant unconnected and again with it connected).

Such a design process does not directly "map onto" the IEC 60880 life cycle and careful interpretation is required to apply particular clauses in the Standard to the relevant part of the development process.

If there are major deviations from the IEC 60880 reference life cycle then the suite of design documentation produced will also vary greatly from that specified in appendix F of the Standard. The Standard is very prescriptive in this area but, providing the information required by the Standard has been documented in the system design documents, no major non-compliance should be raised.

If a generic design, with many existing system components, is to be used as the basis for a new system any new components required may be produced using the original language and development tools. These may not have all of the features identified in section 5.2 of the Standard. The advantages inherent in using a mature system design have to be balanced against the use of less advanced development techniques. Providing the use of mature techniques are adequately justified only a minor non-compliance should be raised (please note that the use of pre-existent software is to be addressed by new IEC 60880 development).

Many clauses in the Standard require careful interpretation when considering modular distributed processing systems containing many subsystems (components), each of which contains a processor. The term "system" as used in the Standard should be interpreted in some instances as applying to the whole distributed system and in others as applying to each of the subsystems.

For example:

Hardware/Software integration may be performed initially on a component by component basis (a component typically being a processor board such as those handling a control system's inputs and outputs). Typically a system contains many subsystems and each contains many identical components. A system change may only affect the internal workings of a component and so, providing the interface between subsystems has not changed, an impact analysis would not identify the need to perform testing beyond a repeat of the component proof test and the subsystem integration testing. Thus, in this case, the subsystem would be considered to be the "system" in IEC 60880, 7.5.

It should be noted that an assessment process may be extremely protracted if the software developed for different subsystems in a system is developed under differing life cycles; in such circumstances it may be advisable to summarize the compliance of different life cycles rather than produce a detailed compliance analysis of each development life cycle against each clause of the Standard.

Thus, to summarize, although many non-compliance findings can be raised because of differences between the recommended life cycle and the actual life cycle (and between the prescribed set of documentation required in the Standard and the actual set of documentation produced) in most instances the principles which the Standard attempts to enforce are clear. Major non-compliance findings should only be raised if there are deviations from these principles.

5.2 Types de logiciels

Les composants logiciels d'un système sont souvent définis soit en tant que logiciel «système» (communications, gestion mémoire, fonctions standard) soit en tant que logiciel d'«application» (logique de verrouillage, boucles d'asservissement, affichages d'images, logique d'alarme, etc.). Le logiciel d'application utilise les ressources fournies par le logiciel système, réduisant ainsi le besoin de duplication du code au sein des modules, diminuant le volume global du logiciel en réduisant par conséquent la possibilité d'erreurs.

Il peut exister deux jeux de documents de conception distincts, un pour le logiciel système «générique», et un pour le logiciel d'application. Le logiciel d'application est généralement spécifique à un projet; le logiciel système peut être utilisé dans plusieurs projets.

L'exactitude des données est généralement considérée comme étant plus facilement vérifiable que celle du code; ainsi, pour tenter d'améliorer l'intégrité et la flexibilité, de nombreuses conceptions de système utilisent largement les données de configuration. Il convient de considérer que cette approche conceptuelle est généralement cohérente avec les principes de la CEI 60880 (voir annexe B, B1.a.a et B2.a.ab). Les données de configuration peuvent être associées au logiciel système ou au logiciel d'application. Les données de configuration comportent généralement des données non modifiables et des données modifiables (c'est-à-dire modifiables en ligne). Les données modifiables comporteraient typiquement des limites d'alarme, des points de consigne, des données requises pour l'étalonnage des instruments, etc.

Ainsi, en résumé, les logiciels correspondant à un système peuvent comprendre les types suivants:

a) logiciel système – existant;

b) logiciel système – à développer;

c) logiciel d'application – à développer;

d) données de configuration – non modifiables, à développer;

e) données de configuration – modifiables, à développer.

Fréquemment, la Norme n'établit pas clairement les distinctions entre ces différents types de logiciels et données, et certaines exigences ne sont pas applicables à toutes les catégories de logiciels. En particulier, lorsque l'on transforme des données de configuration modifiables, de nombreuses exigences de l'article 9 de la Norme ne sont pas applicables, voir commentaires à la fin du paragraphe 5.3 ci-dessous.

Il convient qu'un évaluateur soit expérimenté et connaisse bien les exigences de la CEI 60880 et qu'il soit capable d'interpréter la Norme, en tenant compte des caractéristiques des différents types de logiciels et données au sein d'un système.

5.3 Modifications de logiciel

Dans l'introduction de la CEI 60880, il est spécifié que la Norme contient des recommandations concernant «les procédures relatives à la maintenance du logiciel, la modification, la gestion de la configuration» et que ce sujet est traité à l'article 9 de la Norme. Le champ d'application de cet article est beaucoup plus large que le titre pourrait le laisser penser; en effet, cet article fournit des recommandations relatives aux procédures de gestion des modifications de système.

La Norme impose que chaque demande de modification soit examinée et qu'une analyse des incidences soit effectuée. Le paragraphe 9.2.3 de la Norme fait référence aux articles couvrant la vérification de logiciel et la validation de système programmé; ils doivent être appliqués globalement ou partiellement en fonction des résultats de l'analyse des incidences.

5.2 Software types

The software components of a system are often defined as being either “system” software (communications, memory management, standard functions) or “application” software (interlock logic, control loops, display formats, alarm logic, etc.). Application software uses the facilities provided by the system software, thus reducing the need for duplication of code within modules and thus reducing the overall amount of software and hence the possibility of faults.

There may be two separate sets of design documentation, one for the “generic” system software, and one for the application software. Application software is usually specific to one project; system software may be used in many projects.

The correctness of data is generally considered easier to verify than that of code and in an attempt to improve integrity and flexibility many system designs make extensive use of configuration data. This design approach should be considered to be generally consistent with the principles of IEC 60880 (see appendix B, B1.a.a and B2.a.ab). Configuration data may be associated with the system software or the application software. Configuration data usually consists of unamendable data and amendable data (i.e. amendable on line). Amendable data would typically consist of alarm limits, setpoints, data required to calibrate instrumentation, etc.

So, in summary, software for a system may consist of the following types:

a) system software – existing;

b) system software – to be developed;

c) application software – to be developed;

d) configuration data – non-amendable, to be developed;

e) configuration data – amendable, to be developed.

Often the Standard does not clearly address the distinctions between these various types of software and data and some requirements are not relevant to all categories of software. In particular, when modifying amendable configuration data many of the requirements of clause 9 of the Standard are not applicable, see discussion at the end of subclause 5.3 below.

An assessor should be experienced and knowledgeable about the requirements of IEC 60880 and should be able to interpret the Standard to take account of the characteristics of the different types of software and data within a system.

5.3 Software changes

In the introduction to IEC 60880 it is stated that the Standard contains recommendations for “procedures for software maintenance, modification and configuration control” and this issue is addressed in clause 9 of the Standard. The scope of this clause is much broader than the title would suggest; the clause effectively provides guidelines for system change control procedures.

The Standard requires that each modification request be examined and an impact analysis performed. Subclause 9.2.3 of the Standard references the sections covering software verification and computer system validation and the whole or part of these are to be applied depending on the results of the impact analysis.

Bien que les articles de la Norme relatifs à la gestion des modifications ne reconnaissent pas tous explicitement la nécessité d'introduire des procédures de gestion de façon incrémentielle, comme un système suit les phases du cycle de développement, il convient de considérer une introduction par phase des contrôles comme étant cohérente avec les principes généraux de la CEI 60880. Il convient de prendre en compte le contenu du paragraphe 9.1.2 lorsque l'on envisage l'introduction de contrôles de modification; le paragraphe 9.1.2 recommande d'approuver une demande de modification immédiatement si elle est «mineure» (c'est-à-dire interprétée comme une modification n'affectant pas significativement la conception) et si elle est présentée au cours du développement initial.

Il convient de considérer qu'un processus de développement typique dans lequel sont introduites des procédures de gestion formelles de logiciel – sur une base module par module après la vérification des modules, et avec une mise en oeuvre complète des contrôles de modifications du système (c'est-à-dire l'ensemble des recommandations de la CEI 60880) après le début des tests finals de mise en service sur site – est conforme aux exigences de maîtrise de la configuration fournies au paragraphe 7.3 de la CEI 60880.

Durant le processus de développement, il convient que les utilisateurs évaluent les implications en termes de sûreté des modifications apportées aux exigences fonctionnelles; il n'est pas nécessaire que les utilisateurs examinent les modifications apportées au niveau des moyens/méthodes permettant de remplir ces exigences, et qui sont transparentes pour les utilisateurs, c'est-à-dire que le domaine d'application de la revue devrait être adapté au type de modifications à apporter. Au cours du développement, il n'est pas recommandé de faire évaluer les modifications techniques détaillées, internes aux systèmes programmés, par les personnes responsables des exigences fonctionnelles du système (c'est-à-dire les ingénieurs de la centrale) si ces modifications n'affectent pas ces exigences. Le niveau de revue, ainsi que les personnes réalisant les revues, doivent être adaptés aux types de modifications en cours d'application.

Lorsque l'on considère l'adéquation d'un processus de gestion de modification, il convient de prendre en compte les caractéristiques particulières du logiciel à modifier. Lorsque, par exemple, une conception de système permet la modification de données élémentaires (données de configuration modifiables), la modification de ces données ne nécessite pas d'examen approfondi ni d'analyse d'incidence logiciel, à condition que:

a) la gamme de modifications autorisée soit spécifiée dans les exigences système;

b) la capacité du logiciel à rejeter des valeurs en dehors de l'intervalle autorisé soit vérifiée au cours du processus de conception;

c) la capacité du logiciel à fonctionner correctement sur un intervalle représentatif de valeurs de modification ait été vérifiée.

Le contenu du paragraphe 9.1.2 de la CEI 60880 et le principe d'une application graduelle de la gestion des modifications qu'il implique, doivent guider l'interprétation des articles correspondants de la Norme.

5.4 Fiabilité

La Norme fait largement référence à la spécification des exigences de fiabilité du logiciel et à la vérification de ces exigences, par exemple au niveau de la CEI 60880, paragraphes 4.8, 5.1.2, et particulièrement 6.2.1 qui comporte les termes «évaluer chaque élément du logiciel et pour contrôler à chaque phase le respect des exigences fonctionnelles et de fiabilité».

Ces articles nécessitent une interprétation prudente: en effet, bien que la fiabilité des composants matériels puisse être déterminée par le taux de défaillance mesuré, il n'existe aucun moyen probant pour mesurer la «fiabilité» de la conception d'un logiciel, ni d'un matériel d'ailleurs.

Although not all of the clauses in the Standard which address change control issues explicitly recognize the need to introduce control procedures incrementally, as a system proceeds through the many phases of a development life cycle, a phased introduction of controls should be considered to be consistent with the general principles of IEC 60880. The contents of subclause 9.1.2 should be taken into account when considering the phased introduction of change controls; subclause 9.1.2 states that a change request should be approved immediately if it is "minor" (i.e. interpreted as meaning a change which does not affect the basis of the design) and if it is made during initial development.

A typical development process, where formal software control procedures are introduced, on a module-by-module basis following module verification, with full implementation of system change controls (i.e. the full recommendations of IEC 60880) introduced following the commencement of final site commissioning tests, should be considered to be compliant with the requirements for configuration control provided in subclause 7.3 of IEC 60880.

During the development process changes to the functional requirements should be assessed for safety implications by the users; changes in the means/method of meeting these requirements, and which are transparent to the users, need not be reviewed by them, i.e. the scope of review should be appropriate to the type of changes to be made. During development it is not recommended that detailed technical changes, internal to the software based systems, are assessed by those responsible for the functional requirements of the system (i.e. plant engineers) if such changes do not affect those requirements. The level of review, and those performing reviews, must be appropriate to the types of changes being implemented.

When considering the adequacy of a change control process due account should be taken of the particular characteristics of the software to be changed. Where, for example, a system design allows for the amendment of data items (amendable configuration data), changes to such items do not require extensive review or software impact analysis, providing that:

a) the permitted range of amendment items is specified in the system requirements;

b) the ability of the software to reject values outside the allowed range is verified during the design process;

c) the ability of the software to operate correctly over a representative range of amendment values has been verified.

The contents of IEC 60880, subclause 9.1.2, and the principle of a graded application of change control implied by it, must guide the interpretation of relevant clauses in the Standard.

5.4 Reliability

The Standard makes extensive reference to the specification of software reliability requirements and to the verification of these requirements, e.g. IEC 60880, subclauses 4.8, 5.1.2, and particularly 6.2.1, which contains the phrase "evaluate each item of software and each phase to show whether the functional and reliability requirements specification is met".

Such clauses require careful interpretation for although the reliability of hardware components may be determined by the measured failure rate, there is no proven way of measuring the "reliability" of software, nor indeed hardware, design.

De nombreux argumentaires de sûreté modernes s'appuient sur une analyse probabiliste dans laquelle on attribue une fiabilité nominale à tous les systèmes de sûreté, y compris les systèmes programmés. Plutôt que «fiabilité» (terme plus applicable à des systèmes réalisés par matériel où les composants sont soumis à une dégradation physique au cours du temps), on utilise actuellement en général le terme «niveau d'intégrité» pour les logiciels par exemple, cette approche est utilisée par la CEI 61508.

Le sujet critique à déterminer pour les développeurs et les autorités de sûreté est l'étendue de la démonstration (concernant la qualité du processus d'implémentation du logiciel) appropriée pour s'assurer que l'intégrité du logiciel d'un système est compatible avec la fiabilité déclarée pour le système.

Il n'existe aucun consensus concernant l'étendue de la démonstration appropriée. En effet, il peut être préférable d'exclure la «fiabilité» du logiciel du domaine d'application de l'évaluation. Si la «fiabilité» doit être incluse, on peut s'en remettre à l'expérience et à la compétence de l'évaluateur pour juger de la crédibilité des déclarations qui ont été faites. Parmi les sujets appropriés à considérer figurent:

a) l'expérience opérationnelle;

b) l'étendue des activités de vérification et de validation;

c) la qualité du cycle de développement du logiciel;

d) la conformité générale du processus de développement du logiciel par rapport à la CEI 60880;

e) la possibilité de défaillance de mode commun du logiciel (un futur développement de la CEI 60880 doit fournir un guide sur ce sujet).

De nombreuses caractéristiques de conception de la centrale et du système sont susceptibles de réduire le niveau d'intégrité requis pour un logiciel système, par exemple redondance de systèmes, diagnostics en ligne, redondance de composants du système.

5.5 Indépendance

Le degré d'indépendance requis entre les personnes/organismes réalisant la conception et celles/ceux réalisant les activités de vérification représente un sujet qui concerne plusieurs articles de la Norme. La Norme impose que les personnes réalisant une vérification de module, une vérification de système intégré et une validation de système programmé, soient «indépendantes» voir 6.2.1, 7.5 et l'article 8 de la CEI 60880 et les annexes associées. Le paragraphe 9.2.3 fait également référence à ces articles pour l'implémentation des modifications.

L'indépendance du personnel effectuant la vérification et la validation de logiciel est un sujet de préoccupation général pour toutes les personnes chargées du développement de catégories de logiciels de sûreté, liés à la sûreté et commercialement critiques. Beaucoup de cycles de développement utilisent les techniques suivantes pour introduire un degré d'indépendance dans le processus de vérification:

a) audits d'activités de conception;

b) revue indépendante des propositions de conception, par exemple spécifications de test.

L'utilisation de ces techniques n'est pas directement traitée par la Norme.

La vérification indépendante d'un logiciel est très clairement exigée en 6.2 de la Norme. Pour de nombreux cycles de vie, la seule activité de vérification purement liée au logiciel (d'autres activités de vérification impliquent certains aspects de l'intégration matériel/logiciel) est l'activité de vérification des modules logiciels (généralement pas plus de 100 instructions exécutables). Il convient que la vérification des modules logiciels soit réalisée par un personnel indépendant et il convient de considérer tout manquement à cette exigence de la CEI 60880 comme une non-conformité majeure.

Many modern safety cases are based upon a probabilistic analysis where all safety systems, including software based systems, are allocated a nominal reliability. Rather than “reliability” (a term which is more applicable to hardware-based systems where components are subject to physical degradation over time) the term “integrity level” is now generally used for software, e.g. this approach is taken by IEC 61508.

The critical issue for developers and licensing bodies to determine is what extent of demonstration (of the quality of the software implementation process) is appropriate to be assured that the integrity of a system's software is compatible with the reliability claims made for a system.

There is no consensus as to what extent of demonstration is appropriate. Indeed, it may be appropriate to exclude the consideration of software “reliability” from the scope of an assessment. If “reliability” is to be included then it may be left to the experience and skill of the assessor to judge the credibility of any claims which have been made. Relevant issues to consider include:

a) operational experience;

b) the extent of verification and validation activities;

c) the quality of the software development life cycle;

d) the general compliance of the software development process against IEC 60880;

e) the potential for software Common Mode Failure (a future development of IEC 60880 is to provide guidance on this issue).

There are many plant and system design features which may reduce the integrity level required of a system's software, e.g. redundant systems, on-line diagnostics, redundant system components.

5.5 Independence

The degree of independence required between persons/bodies performing design and those performing verification activities is an issue relevant to many sections of the Standard. The Standard requires that persons performing module verification, integrated system verification and computer system validation, be “independent” see 6.2.1, 7.5, and clause 8 of IEC 60880 and the associated appendices. Subclause 9.2.3 also refers to these clauses for the implementation of changes.

The independence of personnel performing the verification and validation of software is a general issue of concern to all those developing safety, safety-related and commercially critical categories of software. Many development life cycles use the following techniques to introduce a degree of independence into the verification process:

a) audits of design activities;

b) independent review of design submissions, e.g. test specifications.

The use of such techniques is not directly addressed by the Standard.

Independent verification of software is very clearly required in 6.2 of the Standard. For many development life cycles the only purely software verification activity (other verification activities involving some aspect of hardware/software integration) is that of software module (usually no more than 100 executable statements) verification. Software module verification should be performed by independent personnel and failure to address this requirement of IEC 60880 should be considered to be a major non-compliance.

Il peut ne pas être pratique de mettre en oeuvre la totalité des exigences de 6.2.1 de la CEI 60880 concernant l'indépendance pour toutes les activités de vérification et validation. Cependant, il convient de soumettre toutes ces activités à une revue/un audit indépendant (techniques a) et b) ci-dessus). A condition que l'intégrité du processus ne soit pas compromise, il convient d'avoir recours à l'article «Si des pratiques différentes de celles figurant dans les annexes sont utilisées, elles doivent être documentées et pouvoir être vérifiées...» à l'article 1 de la CEI 60880.

Par exemple, les spécifications de test d'intégration – elles peuvent être produites par l'équipe de conception, puis comparées aux exigences définies pour le système par un autre organisme (un organisme «indépendant» non impliqué dans la conception détaillée du système – voir article 8, avant-dernier alinéa de la page 32). Dans ce cas, bien que la Norme n'ait pas été suivie «à la lettre», et à condition qu'une revue rigoureuse ait été effectuée, le principe d'indépendance a été respecté; si le processus est contrôlé et peut être vérifié, alors il convient de ne prononcer qu'une non-conformité mineure.

6 Interprétation des articles spécifiques de la CEI 60880

Une sélection d'articles de la CEI 60880 fait l'objet de commentaires ci-dessous (il est à noter que les numéros de référence et le texte de la CEI 60880 sont imprimés en italique).

Référence CEI.

3 Structure du projet

3.1.6 Chaque phase doit systématiquement se terminer par une revue critique. Les revues critiques constituent une part importante du processus de vérification du logiciel (voir article 6).

Un travail en parallèle sur les différentes phases du cycle de développement peut être acceptable sous réserve que les principes fondamentaux de la Norme soient pris en compte, à savoir que le système final est une réalisation des exigences de l'utilisateur et que les processus de vérification/validation sont détaillés, complets, maîtrisés et auditables. Une phase de développement peut s'achever par une revue, mais du fait de la nature itérative du processus de conception, chaque phase sera exécutée plusieurs fois.

NOTE 1 – L'ISO 9001 impose également la réalisation de revues mais n'exclut pas l'utilisation d'un cycle de vie itératif.

NOTE 2 – La CEI 61508 reconnaît également la nature itérative du processus de développement logiciel.

3.1.7 Chaque vérification ou revue critique doit donner lieu à un rapport sur les analyses réalisées, les conclusions et les décisions acceptées. Ce rapport doit être inclus dans la documentation.

Le terme «rapport» peut être interprété comme une «consignation», c'est-à-dire qu'il faut une consignation documentée des activités de chaque étape de vérification et de chaque revue critique.

3.1.8 Une liste des documents requis au minimum durant la vie du logiciel est donnée dans l'annexe F.

Compte tenu des différences importantes existant entre les méthodes de conception et les nombreuses relations contractuelles qui peuvent exister entre les sociétés impliquées dans le processus de développement, l'utilisation de la liste documentaire fournie dans l'annexe F peut être non réalisable. Cette liste devrait être considérée comme indicative et aucune non-conformité majeure ne devrait être prononcée sur le processus de développement, sous réserve qu'un ensemble détaillé et complet de documents soit produit, indépendamment du format ou du contenu d'un document particulier.

It may not be practical to implement the full requirements of 6.2.1 of IEC 60880 for independence for all verification and validation activities. However, all such activities should be subject to independent review/audit (techniques a) and b) above). Providing the integrity of the process is not compromised then the clause “If practices differing from those of the appendices are used they shall be documented and auditable.” in clause 1 of IEC 60880 should be invoked.

For example, the production of integration test specifications – these may be produced by the design team and then reviewed against the defined system requirements by another body (an “independent” body which has not been involved in the detailed design of the system – see clause 8, penultimate paragraph of page 33). In this case, although the Standard has not been followed "to the letter”, providing a rigorous review has been performed, then the principle of independence has been addressed; if the process is controlled and auditable, then only a minor non-compliance should be raised.

6 Interpretation of specific clauses of IEC 60880

A selection of IEC 60880 clauses are discussed below (note that IEC 60880 reference numbers and text are shown in italics).

IEC reference

3 Project structure

3.1.6 Each phase shall be systematically terminated by a critical review. Critical reviews form a major part of the verification process of the software project (see clause 6).

Parallel working on the different phases of the development life cycle may be acceptable providing that the fundamental principles of the Standard are addressed, namely that the final system is a realisation of the users' requirements and that the verification/validation processes are comprehensive, controlled and auditable. A development phase may be “terminated” by a review, but due to the iterative nature of the design process each phase will be revisited many times.

NOTE 1 – ISO 9001 also requires that design reviews be held but does not exclude the use of an iterative life cycle.

NOTE 2 – IEC 61508 recognizes the iterative nature of the software development process.

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

The term “report” may be interpreted to mean “record”, i.e. there must be a documented record of each verification step and critical review.

3.1.8 A list of documents required as a minimum through the software life cycle is given in appendix F.

Given the wide variation between different design methodologies and the many contractual relationships which can exist between companies involved in the development process, it may not be practical to use the documentation list provided in appendix F. This list should be considered to be for guidance purposes only and no major non-compliance should be raised against a development process providing a comprehensive set of documents is produced, irrespective of the format and contents of any particular documentation.

Référence CEI

4 Spécifications du logiciel

4.1 Les spécifications du système programmé doivent donner une vue externe et synthétique des fonctions qui doivent être réalisées par le logiciel du système programmé (A2.1).

Dans ce contexte, le terme «fonctions» signifie fonctions d'applications (c'est-à-dire relatives à la centrale) telles que fermeture de vannes, verrouillages, consignes à l'opérateur, etc. Le terme «synthétique» dans ce contexte fait référence à des techniques telles que diagrammes d'état/transition, diagrammes logiques, etc. De telles techniques sont applicables à de nombreux aspects des exigences du logiciel d'application et ont l'avantage d'être facilement comprises par les ingénieurs de la centrale et les analystes logiciels. Elles peuvent ne pas être applicables à toutes les exigences fonctionnelles par exemple une exigence pour l'affichage de l'heure sur des écrans peut ne pas être représentable de façon «synthétique». Un système devrait être considéré conforme sous réserve qu'une méthode appropriée ait été utilisée pour définir les fonctions demandées.

4.2 La configuration du système programmé doit être décrite en prenant en compte ses exigences de fiabilité et son environnement, étant donné que les caractéristiques de fiabilité et la configuration du système sont intimement liées (A2.2).

La fiabilité d'une conception logicielle est traitée au paragraphe 5.4 de ce rapport.

4.3 Les principes du Guide 50-SG-D3 de l'AIEA, et spécialement ceux figurant aux paragraphes 7.12 et 7.3.2, sont applicables en ce qui concerne l'interaction entre le système programmé et l'opérateur ou toute autre personne.

Quand on définit le domaine d'une évaluation, il convient d'examiner avec attention si le contenu du guide AIEA 50-SG-D3 doit être inclus ou non. Une évaluation détaillée par rapport au guide AIEA peut se révéler peu pratique du fait du temps nécessaire et des différentes compétences requises pour évaluer les sujets relatifs à l'ergonomie, par exemple le bénéfice escompté peut ne pas justifier l'investissement en temps et en compétences.

4.6 Les contraintes entre matériel et logiciel doivent être décrites (A2.6). La référence au document de spécifications du matériel doit être indiquée.

Des contraintes peuvent être identifiées par référence aux manuels appropriés du matériel.

4.7 Les conditions particulières de fonctionnement, tels que le démarrage de la centrale et le rechargement en combustible, doivent être décrites (A2.7).

Pour les systèmes qui n'ont pas à prendre en compte des conditions particulières de fonctionnement, une réponse «non applicable» est appropriée.

4.8 Le logiciel du système programmé doit posséder des fonctions d'autocontrôle couvrant le logiciel lui-même et le matériel hôte (A2.8). Cela est considéré comme un facteur essentiel pour atteindre l'objectif de fiabilité de l'ensemble du système.

Une interprétation de l'application de cette clause est nécessaire dans une certaine mesure car il n'est pas possible pour le diagnostic d'un système d'identifier 100 % des pannes potentielles du système. Il est possible que la fiabilité demandée puisse être atteinte sans diagnostics en ligne intensifs par exemple par une redondance multiple et des essais en ligne fréquents. Toute redondance du système doit être bien prise en compte lors de l'évaluation de la conformité d'un système avec cet article.

IEC reference

4 Software requirements

4.1 The computer system specification shall give an external and synthetic view of the functions to be implemented by the software of the computer system (A2.1).

The term ”functions” in this context means applications (i.e. plant related) functions such as closing valves, interlocks, driving operator indications, etc. The term “synthetic” in this context refers to techniques such as state transition diagrams, logic diagrams, etc. Such techniques are applicable to many aspects of the requirements of application software and have the advantage that they are easily understood by both the plant engineers and the software analysts. It may not be practical to apply such techniques to all functional requirements e.g. a requirement to display the time on VDU formats could not be “synthetically” represented. A system should be considered compliant providing an appropriate method has been used to define the functions required.

4.2 The computer system configuration shall be described using as a background the reliability requirements and the environment of the system, since reliability properties and system configuration are closely connected (A2.2).

Software design reliability is discussed in section 5.4 of this report.

4.3 For the interaction between the computer system and the plant operator or any other person, the principles of IAEA “Guide 50-SG-D3” in general and especially those of subclauses 7.12 and 7.3.2 are applicable.

When defining the scope of an assessment, careful consideration should be given as to whether or not the contents of IAEA “Guide 50-SG-D3” are to be included. It may not be practical to perform a detailed assessment against the IAEA guide due to the time involved and the different skills required to assess ergonomic issues, i.e. the benefit to be gained may not warrant the investment in time and skilled effort.

4.6 The constraints between hardware and software shall be described (A2.6). A reference to the hardware requirements document shall be made.

Constraints may be identified by reference to appropriate hardware product manuals.

4.7 Special operating conditions such as plant commissioning and refuelling shall be described (A2.7).

For systems which do not have a requirement to support special operating conditions a response of “not applicable” is appropriate.

4.8 The computer system software shall continuously supervise both itself and the hardware (A2.8). This is considered a primary factor in the achievement of the overall system reliability requirements.

Some interpretation of the application of this clause is required as it is not possible for a system's diagnostics to identify 100 % of potential system faults. It is possible that the required reliability may be attained without extensive on-line diagnostics, e.g. through multiple redundancy and frequent on-line test. Due account of any system redundancy should be taken when assessing the compliance of a system against this clause.

Référence CEI

4.8.1 L'autocontrôle doit remplir les exigences suivantes, à moins qu'il ne puisse être prouvé que d'autres moyens procurent le même degré de sûreté:

a) aucune panne unique ne doit pouvoir bloquer directement ou indirectement une fonction quelconque destinée à prévenir un rejet de radioactivité;

Ce paragraphe exige qu'aucune panne de composant unique ne provoque de blocage directement ou indirectement, etc. Il est possible de satisfaire cette exigence par des moyens logiciels ou matériels. Il convient de tenir compte de toute redondance au sein des systèmes ou entre les systèmes.

b) les zones mémoires qui contiennent les paramètres ou les constantes doivent être protégées en écriture de manière à prévenir tout changement de leur contenu.

Plutôt que de contrôler fréquemment une mémoire volatile, il peut être approprié d'utiliser une mémoire non volatile (telle qu'une mémoire EPROM pour contenir les constantes) nécessitant des vérifications moins fréquentes.

4.8.4 Les autocontrôles du système ne doivent pas dégrader les fonctions qu'il doit remplir.

Certaines diminutions des performances d'un système dues à des autocontrôles sont parfois inévitables. Il convient que toute diminution de performance soit quantifiable et que le système continue de respecter ses prescriptions de conception.

4.8.5 Le logiciel du système de sûreté doit être conçu pour satisfaire aux exigences du test périodique qui a lieu à des intervalles de temps maximaux spécifiés (par exemple pendant les périodes d'arrêt).

a) Toutes les fonctions élémentaires doivent être couvertes par le test périodique.

L'utilisation du terme «fonction» dans ce contexte nécessite une interprétation; le terme fait référence à des fonctions d'application (fonctionnement d'une vanne, d'un commutateur, etc.) exprimées dans les exigences du système. L'article ne fait pas référence à des fonctions logicielles système/noyau.

b) Toute dégradation dans l'exécution des fonctions de sûreté doit être détectée.

Il est nécessaire d'effectuer un test pour confirmer que les résultats des actions de sûreté correspondent à la spécification (qui inclurait un facteur de sécurité).

c) Les autocontrôles de base doivent être vérifiables durant les tests périodiques.

Il convient que le logiciel soit capable d'acquérir automatiquement toutes les informations obtenues durant le test périodique. Voir le paragraphe 10.3 pour d'autres détails sur le test périodique.

Il est possible que la fonction historique soit fournie par l'équipement de test hors ligne – ce qui est en fait généralement considéré comme préférable puisqu'il convient que le logiciel en ligne soit aussi simple que possible.

4.11 Les exigences fonctionnelles pour le logiciel (analyse fonctionnelle) doivent être fournies:

– *aux auteurs du document spécifiant les exigences de l'ensemble du système programmé, pour évaluation et accord avant programmation;*

Voir commentaires généraux sur les cycles de développement en 5.1.

IEC reference

4.8.1 Self-supervision shall meet the following requirements, unless it is proved that other means provide the same degree of safety:

a) no single failure shall be able to block directly or indirectly any function dedicated to the prevention of the release of radioactivity;

This subclause requires that any single component failure does not block directly or indirectly, etc. This requirement may be achieved by either software or hardware means. Due account should be taken of any redundancy within systems and between systems.

b) those parts of the memory that contain code or invariable data shall be monitored to prevent any changes.

Rather than frequently monitor volatile memory it may be appropriate to use non-volatile memory (such as EPROMs to hold invariable data) which would require less frequent checking.

4.8.4 System self-checking shall not adversely affect the intended system functions.

Some reduction of a system's performance due to self-checks is sometimes unavoidable. Any reduction in performance should be quantifiable and the system should continue to meet its design requirements.

4.8.5 The safety system 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).

a) Every single function shall be coverable by periodic testing.

The use of the term “function” in this context requires some interpretation; the term refers to application functions (operation of valve, switch, etc.) as expressed in the system requirements. The clause does not refer to system/kernel software functions.

b) Any degradation of the execution of safety functions shall be detected.

A test to confirm that the performance of all safety actions is within specification (which would include a safety factor) is required.

c) The basic self-checking functions shall also be testable during periodic tests.

The software should be able to collect automatically all information gained during periodic testing. Further details for periodic testing are given in subclause 10.3.

The logging function may be provided by the off-line test equipment – indeed this is generally considered preferable as the on-line software should be as simple as possible.

4.11 The software functional requirements shall be provided:

– *to the authors of the computer system requirements documents for assessment and approval prior to programming;*

See general comments on development life cycles in 5.1.

Référence CEI

– *aux responsables de la sûreté pour autorisation d'exploitation, en ce qui concerne le respect de l'ensemble des exigences de sûreté du réacteur.*

Les responsables de la sûreté ont souvent l'option d'examiner tout document de conception et la sélection de la documentation est laissée à leur discrétion. Il est souvent plus commode de produire un jeu de documents de sûreté distinct spécifiant toutes les exigences du système appropriées (c'est-à-dire pour le processus de sûreté).

5 Développement (conception et codage) du logiciel des systèmes de sûreté

Il convient que la spécification des exigences fonctionnelles à remplir par le logiciel soit disponible avant les phases de conception et de codage du développement des programmes.

Voir commentaires généraux sur le cycle de développement en 5.1.

Voir également les commentaires sur le paragraphe 4.11.

5.1.1 Les principes généraux de développement ci-après sont basés sur l'expérience dans la production de logiciels exempts d'erreur et compréhensibles.

– *La conception du logiciel doit inclure l'autosurveillance des flux de contrôle et de données. La détection d'une erreur doit entraîner l'action appropriée, conformément au paragraphe 4.8.*

Ce que la Norme exige en la matière n'est pas très clair, mais un moyen d'effectuer une autosurveillance du flux de contrôle consiste à utiliser un chien de garde, c'est-à-dire que le logiciel met régulièrement à jour une adresse en mémoire, ou fixe un résultat qui est vérifié indépendamment; si le logiciel ne reboucle plus, cette erreur est détectée.

– *Il convient que la structure du programme soit simple et facile à comprendre, à la fois dans sa conception générale et dans ses détails. Il est recommandé de bannir les «astuces», les structures récursives et la réduction de code non nécessaires.*

Le terme «astuces» fait référence aux techniques employées par le programmeur et qui réduisent la maintenabilité et la lisibilité du code. Il convient qu'une norme de codage approuvée interdise ces techniques. Cependant, il est habituellement possible de justifier l'utilisation de certaines techniques de codage, telles que l'utilisation d'équivalences pour accéder à un tampon de données par octet ou par mot quand le code qui en résulte est simplifié et la lisibilité améliorée.

5.1.2 Les recommandations suivantes dérivent de ces principes.

Il convient que:

a) les mesures à prendre pour obtenir la fiabilité requise, y compris l'autocontrôle, soient choisies au début du développement (B3):

Voir 5.4 du présent rapport technique.

b) une approche descendante soit préférée à une approche ascendante dans le développement du logiciel (B1):

Voir commentaires généraux sur le développement descendant, le travail en parallèle et l'utilisation de composants préexistants à l'article 5.

c) un modèle conceptuel de la structure du système soit adopté au début de chaque projet logiciel (B2):

Il convient que le modèle conceptuel soit documenté.

IEC reference

– *to the licensers with respect to compliance with the overall plant safety requirements for licensing approval.*

The licensing authority usually have the option of examining any design document and the selection of documentation would be at their discretion. It is often more convenient to produce a separate set of licensing documents which specify all relevant (i.e. to the licensing process) system requirements.

5 Development (design and coding) of safety system software

The software functional requirements specification should be available, before the design and coding phase of program development begins.

See general comments on the development life cycle in 5.1.

Also see comments on subclause 4.11.

5.1.1 The following general principles for development are based on experience in producing error-free and understandable software.

– *The software design shall include self-supervision of control flow and data. On failure detection, appropriate action shall be taken in accordance with subclause 4.8.*

It is not clear what that Standard requires in this regard, but a means of providing self-supervision of control flow is to use a watchdog i.e. the software regularly updates a memory location, or sets an output, which is independently checked; if the software fails to loop this error will be detected.

– *The program structure should be simple and easy to understand, both in its overall design and in its details. Tricks, recursive structures and unnecessary code compaction should be avoided.*

The term “tricks” refers to techniques which are employed for the convenience of the programmer and which detract from code maintainability and legibility. An approved coding Standard should proscribe such techniques. However, it is usually possible to justify the use of some special coding techniques, such as using overlays to enable a data buffer to be referenced by byte or by word, where the resulting code is simplified and legibility improved.

5.1.2 From these principles the following recommendations are derived:

a) measures to obtain the required reliability including self-supervision should be chosen at the outset of the development (B3):

See 5.4 of this technical report.

b) a top down approach to software development should be preferred to a bottom up approach (B1):

See general comments on top-down development, parallel working and use of pre-existent components in clause 5.

c) a conceptual model of system structure should be adopted at the beginning of each software project (B2):

The conceptual model should be documented.

Référence CEI

e) dans le cas d'utilisation d'un logiciel commercialisé, la démonstration de son bon fonctionnement soit réalisée (B2.c).

La justification peut se fonder sur l'expérience opérationnelle (l'étendue de la base utilisateur) et sur l'étendue des tests logiciels. Lorsque l'on peut montrer que la qualité des processus de développement pour le logiciel commercialisé est équivalente à celle requise par la CEI 60880, il n'est pas obligatoire de fournir des enregistrements opérationnels complets.

NOTE – Un nouveau développement de la CEI 60880 fournira des recommandations supplémentaires concernant l'utilisation de logiciels préexistants.

5.2 Langage, traducteur et autres outils

5.2.1 Il convient d'utiliser des langages disposant d'un traducteur complètement testé. Si un traducteur partiellement testé est utilisé, une vérification supplémentaire devra montrer que le résultat de la traduction est correct.

Il est généralement admis d'utiliser un outil de traduction commercialement disponible à condition que le résultat de la traduction soit correctement testé et que l'outil ait un retour d'expérience acceptable (l'étendue des tests requis peut varier en fonction de la maturité/qualité de l'outil).

Lorsque la qualité d'un compilateur ne peut pas être validée de façon adéquate et quand il existe des exigences d'intégrité logicielle très importantes, l'utilisation d'une technique telle que la décompilation (Comparateur de Source du Code) peut être nécessaire.

5.2.2 Il convient que le langage soit défini de manière complète et non ambiguë, à défaut, son emploi devra être limité aux seuls éléments définis de manière complète et non ambiguë. Cela s'applique également quand il existe un doute dans la traduction d'un élément particulier du langage ou une combinaison particulière d'éléments.

Pour un langage industriel normalisé, tel que le langage « C », certaines caractéristiques sont généralement appliquées différemment par différents fournisseurs (en attendant qu'une norme internationale fasse l'objet d'un accord, ce qui peut prendre 10 ans pour un nouveau langage). Cependant, pour tout compilateur particulier, l'application est cohérente et il convient de la documenter entièrement dans le guide d'utilisation. De nombreux systèmes sont développés en utilisant des langages propriétaires pour lesquels l'interprétation sémantique «standard» n'intervient pas puisque le fournisseur définit et documente le produit. Le principe à suivre consiste, pour les programmeurs, à ne pas utiliser de caractéristiques de compilateur non documentées.

5.2.4 De même que les points spécifiques mentionnés dans l'annexe D, il est recommandé qu'un langage de programmation d'un système de sûreté et son traducteur n'interdisent pas par leur conception:

– *les constructions limitant les erreurs;*
– *la vérification des types lors de la traduction;*
– *la vérification des types et des limites de validité des pointeurs de tables, ainsi que la vérification des paramètres lors de l'exécution.*

Il convient que les évaluateurs reconnaissent qu'il peut ne pas être possible de fournir ces moyens si l'on utilise une technique (ancienne) stabilisée, éprouvée et fiable. De même, l'exigence relative à la vérification des types et des limites de validité des pointeurs de tables, ainsi que la vérification des paramètres lors de l'exécution peut nécessiter des ressources processeur excessives pour certaines applications.

IEC reference

e) where standard software from a manufacturer or supplier is used, it should be shown to have operated satisfactorily (B2.c).

Justification may be based upon operational experience (the extent of the user base) and the extent of software testing. Where the quality of the development processes for standard software can be shown to be equivalent to that required by IEC 60880 then there is no requirement for extensive operational records to be made available.

NOTE – New development of IEC 60880 is to provide further guidance on the use of pre-existing software.

5.2 Language, translator and other tools

5.2.1 Languages with a thoroughly tested translator should be used. If no thoroughly tested translator is employed, additional verification shall show that the result of the translation is correct.

It is generally permissible to use a commercially available translator tool providing the output of the tool is adequately tested and the tool has an acceptable pedigree (the extent of testing required may vary according to the maturity/quality of the tool).

Where the quality of a compiler cannot be adequately validated and where there are very high software integrity requirements, then the use of a technique such as a back compiler (Source to Code Comparator) may be necessary.

5.2.2 The language should be completely and unambiguously defined, otherwise the use of the language shall be restricted to completely and unambiguously defined features. This applies in a similar way if there is any doubt about the correct translation of a specific language feature or a particular combination of such features.

For an industrial standard language, such as "C", there are usually some features which have been implemented differently by different suppliers (until an international Standard is agreed, which may take 10 years for a new language). However, for any particular compiler the implementation is consistent and should be fully documented in the user-guide. Many systems are developed using proprietary languages where the "Standard" interpretation of semantics is not an issue as the supplier defines and documents the product. The principle which must be followed is that programmers shall not use undocumented compiler features.

5.2.4 As well as the specific points mentioned in appendix D, a programming language for safety systems and its translator should not prevent by their design:

– *error-limiting constructs;*
– *translation-time type checking;*
– *run-time type and array bound check, and parameter checking.*

It should be recognized by assessors that it may not be practical to provide these facilities if a mature, tried and trusted (old) implementation technology is used. Also, the requirement for run-time type and array bound check, and parameter checking may require excessive processor resource for some applications.

Référence CEI

5.2.5 Il convient que des outils automatiques de test soient disponibles.

Bien que l'utilisation de tels outils soit recommandée, il s'agit plus d'une question économique que d'une question de sûreté et il convient de juger toute non-conformité à ce sujet en conséquence.

NOTE – La révision de la CEI 60880 fournira des recommandations supplémentaires concernant l'utilisation des outils.

5.3.2 Utilisation des recommandations

Il convient que les recommandations applicables au programme à développer soient choisies et notées dès le début de tout projet de réalisation de logiciel concernant un système de sûreté. Ce choix peut inclure une modification dans l'ordre des priorités des recommandations ainsi qu'une subdivision en détail des recommandations individuelles.

Lorsqu'une évaluation est réalisée rétrospectivement, sur un système préexistant, cette exigence visant à effectuer une pré-évaluation par rapport à la CEI 60880 n'a que peu d'importance, étant donné qu'un évaluateur dispose de tous les éléments pour réaliser une évaluation complète par rapport au cycle de développement réel. Dans ce type de cas, l'application de la mention «non applicable» au niveau de cet article est appropriée.

5.4.1 Lors du développement du programme, la fin de la phase de conception doit donner lieu à un document formalisé, l'analyse organique (F.M3). Ce document est utilisé comme référence pour la revue d'étude de conception et le codage ultérieur.

Des détails suffisants doivent y être inclus pour permettre le codage sans clarification supplémentaire. Il est recommandé de structurer le document pour permettre son développement parallèlement au processus de conception. Un résumé du contenu proposé est donné dans l'annexe C.

Le terme «analyse organique» est obsolète et le nouveau terme équivalent est «exigences de conception du logiciel».

5.4.4 Il est recommandé de choisir la présentation de la documentation selon le besoin particulier, incluant:

- *la description narrative;*
- *les expressions arithmétiques;*
- *toute forme de diagrammes et dessins.*

En règle générale, il est préférable de choisir une représentation graphique. La documentation elle-même doit suivre les normes nationales.

A condition que la documentation système ait un format cohérent, qu'elle soit claire et intelligible, on peut affirmer la conformité.

6 Vérification

6.1 Processus de vérification

Voir les commentaires généraux de l'article 5 sur la nature itérative du cycle de développement.

IEC reference

5.2.5 Automatic testing aids should be available.

Although the use of such tools is recommended this is more an economic issue than one of safety and any non-compliance regarding this issue should be judged accordingly.

NOTE – The revision of IEC 60880 is to provide further guidance on the use of tools.

5.3.2 Use of the recommendations

At the outset of any safety related programming project those recommendations should be selected and recorded which apply to the intended program development. The selection may include the change of priorities and further subdivide into details of individual recommendations.

Where an assessment is performed retrospectively, on a pre-existent system, this requirement to perform a pre-assessment against IEC 60880 is of little value as all the evidence is available for an assessor to perform a full assessment against the actual software life cycle. In such cases an entry of “not applicable” for this clause is appropriate.

5.4.1 During program development the end of the design phase shall be marked by production of a formal document, the software performance specification (F.M3). This document serves as the basis for the formal design review and the subsequent program coding.

Sufficient details shall be included so that program coding can proceed without further design clarification. The document should be structured for continued expansion in parallel to the design process. An outline of the suggested content is given in appendix C.

The term “software performance specification” is now obsolete and the equivalent modern term is “software design specification”.

5.4.4 Documentation formats should be selected according to the specific purpose, including:

– *narrative description;*
– *arithmetic expressions;*
– *appropriate diagrams and drawings.*

As a general rule it is preferable to choose a graphical representation. The documentation itself shall follow national Standards.

Providing the system documentation is to a consistent format, and is clear and intelligible, compliance may be claimed.

6 Verification

6.1 Software verification process

See general comments in clause 5 on the iterative nature of the development life cycle.

Référence CEI

6.2.1 Plan de vérification

En parallèle avec les phases de développement du logiciel décrites ci-dessus, un plan de vérification doit être établi. Ce plan doit expliciter tous les critères, les techniques et les outils à utiliser dans le processus de vérification. Il doit décrire les activités à réaliser pour évaluer chaque élément du logiciel et pour contrôler à chaque phase le respect des exigences fonctionnelles et de fiabilité. Le niveau de détail doit être tel qu'un groupe indépendant puisse exécuter ce plan de vérification et émettre un jugement objectif sur la capacité ou non du logiciel à satisfaire aux exigences de performances.

NOTE – L'indépendance implique que la vérification soit effectuée par un individu ou une organisation séparée de l'individu ou de l'organisation réalisant le développement. La meilleure manière est la mise en place d'une équipe de vérification.

Il n'existe aucun moyen probant de démontrer la fiabilité d'un logiciel – voir 5.4.

Il est important que les vérificateurs soient indépendants mais il est aussi important qu'ils soient compétents. Plusieurs organisations considèrent qu'une période de travail en tant que vérificateur constitue une importante partie du processus de formation des ingénieurs en logiciel et vice versa. Il est possible de juger acceptable que des ingénieurs travaillent sur le développement et la vérification du code pour un projet particulier (en particulier pour les grands projets qui durent plusieurs années et qui peuvent consommer une large proportion des ressources d'une société), à condition que les ingénieurs ne réalisent pas individuellement de vérification sur les modules ou sous-systèmes dont ils étaient responsables auparavant.

L'indépendance doit être assurée en fournissant une structure de gestion dans laquelle le responsable de la vérification dispose d'une marge de manoeuvre suffisante pour réaliser son activité (comme le prescrit la Norme). Il convient de documenter le processus de vérification et de le soumettre régulièrement à un audit indépendant.

Il peut être acceptable que certaines activités de vérification, telles que la production de documents de conception, fassent l'objet d'une évaluation par les pairs au cours du processus de développement à condition qu'une vérification du code indépendante et auditable en regard des spécifications ait lieu avant de placer le système en service actif.

e) l'évaluation des résultats de la vérification obtenus directement à partir des outils de vérification et des tests, évaluation du respect des exigences de fiabilité.

Il convient d'évaluer la qualité du système au vu des erreurs de développement détectées lors de la vérification et de la validation. La fiabilité des composants matériels peut être déterminée en fonction du taux de défaillance mesuré mais il n'existe aucun moyen probant de mesurer la «fiabilité» de conception d'un logiciel ou d'un matériel, voir 5.4, et il convient de ne prononcer aucune non-conformité majeure par rapport à l'article pour cette raison.

6.2.2 Vérification de la conception, revues critiques

c) le respect des exigences de qualité.

Le résultat de la vérification de la conception doit être explicité dans un rapport (F.M5).

Ce rapport doit indiquer:

– *les éléments qui ne sont pas conformes aux exigences fonctionnelles du logiciel;*
– *les éléments qui ne sont pas conformes aux normes de conception;*
– *les modules, données, structures et algorithmes peu adaptés au problème.*

Lorsque l'on envisage le processus de vérification pour de grands systèmes, il n'est pas pratique d'inclure toutes les conclusions de la vérification dans un seul rapport final. Il est couramment acceptable de présenter et de résoudre les points individuels sur des modules particuliers dans le cadre d'un processus progressif et auditable.

NOTE – Dans ce paragraphe de la Norme, la référence à F.M5 est incorrecte, la bonne référence est F.M3.

IEC reference

6.2.1 Verification plan

Concurrently with the phases of the software development cycle described above a software verification plan shall be established. The plan shall document all the criteria, the techniques and tools to be utilized in the verification process. It shall describe the activities to be performed to evaluate each item of software and each phase to show whether the functional and reliability requirements specification is met. The level of detail shall be such that an independent group can execute the verification plan and reach an objective judgement on whether or not the software meets its performance requirements.

NOTE – The requirements for an independent group implies verification either by an individual or an organization which is separate from the individual or organization developing the software. The most appropriate way is to engage a verification team.

There is no proven way of demonstrating the reliability of software – see 5.4.

It is important that verifiers are independent, but it is also important that they are competent. Many organizations consider a period of work as a verifier to be an important part of the training process for software engineers and vice versa. It may be considered acceptable for engineers to work on code development and verification for a particular project (especially for large projects which last for many years and which may consume a large proportion of the resources of one company), providing individual engineers do not perform the verification of modules or subsystems for which they were previously responsible.

Independence must be ensured by providing a management structure where the verification manager has sufficient discretion to perform this activity (as required by the Standard). The verification process should be documented and subject to regular independent audit.

It may be acceptable for some verification activities, such as the production of design documents, to be verified by peer review during the development process providing that there is an independent and auditable verification of code against specifications before the system is placed in active service.

e) evaluation of verification results gained from verification equipment directly and from tests, evaluation of whether the reliability requirements are met.

The quality of the system should be assessed in the light of the implementation errors detected by verification and validation. The reliability of hardware components may be determined by the measured failure rate but there is no proven way of measuring the “reliability” of a software or hardware design, see 5.4, and no major non-compliance should be raised against the clause for this reason.

6.2.2 Design verification, critical reviews

c) the respect of quality requirements.

The result of the design verification shall be documented in a report (F.M5).

This report shall include:

– *items which do not conform to the software functional requirements;*

– *items which do not conform to the design standards;*

– *modules, data, structures and algorithms poorly adapted to the problem.*

When considering the verification process for large systems it is not practical for all verification findings to be contained in one final report. It is usually acceptable for the individual findings on particular modules to be raised and resolved in a progressive and auditable process.

NOTE – The reference in this subclause of the Standard to F.M5 in incorrect; the correct reference is F.M3.

Référence CEI

6.2.3 Un test d'intégration des modules doit alors être effectué pour montrer au plus tôt l'interaction correcte de tous les modules pour une fonction donnée.

Les tests d'intégration doivent démontrer que les modules matériels et logiciels du système fournissent ensemble la fonctionnalité requise. Il peut exister plusieurs phases de tests d'intégration et un processus ascendant et progressif s'avère généralement efficace.

La spécification de test du logiciel constitue des directives pour les activités de vérification du code.

Des recommandations générales relatives aux activités de vérification du code peuvent être fournies dans un document générique, tel que le plan de vérification de logiciel (et procédures associées); il convient de fournir des recommandations spécifiques pour les tests individuels dans les spécifications de test, si nécessaire.

6.2.3.1 Spécification de test du logiciel

La spécification de test du logiciel s'appuie sur le descriptif général du logiciel à tester. Cette description est fournie par l'équipe de développement.

Cet article traite des tests du logiciel; cependant, dans plusieurs cas de cycles de développement, il se peut que le seul test de logiciel soit le test de module du logiciel (à l'issue de ce test, le logiciel peut être testé en association avec le matériel d'exécution, c'est-à-dire selon des tests d'intégration).

La spécification de test du logiciel doit inclure:

a) l'environnement dans lequel s'effectue le test;

b) les procédures de test;

c) les critères d'acceptation, c'est-à-dire la définition détaillée des critères à remplir pour accepter les modules et les composants importants du logiciel au niveau sous-système et système;

d) la détection d'erreurs et les procédures de correction;

e) une liste de tous les documents qu'il est recommandé de produire lors de la phase de vérification du codage.

Il convient qu'aucune non-conformité ne soit prononcée en relation avec cet article à condition que les points a) à e) soient spécifiés à un endroit adéquat dans la hiérarchie documentaire.

6.2.3.2 Compte rendu de test du logiciel

Il doit présenter les résultats du programme de test décrit dans la spécification de test du logiciel en indiquant s'il satisfait ou non aux exigences données dans l'analyse fonctionnelle. Ce document (F.M4) doit permettre un diagnostic complet des anomalies de conception et de fonctionnement. Il doit aussi comprendre la correction des anomalies et des divergences, entre résultats attendus et résultats obtenus, découvertes lors du test.

Le compte rendu de test du logiciel doit inclure les rubriques suivantes à la fois au niveau du module et des niveaux plus élevés:

A condition que toutes les informations ci-dessus soient documentées, il n'est pas nécessaire qu'elles soient toutes contenues dans un rapport unique.

a) configuration du matériel utilisé pour le test;

Il est admis que la configuration soit documentée dans la spécification de test (ou tout autre document approprié) à condition que la configuration soit vérifiée avant de réaliser les tests.

IEC reference

6.2.3 A module integration test shall be performed to show at the earliest stage of development that all modules interact correctly to perform the intended function.

Integration tests must demonstrate that the system hardware and software modules together provide the required functionality. There may be many phases of integration testing and a bottom-up, progressive process is generally found to be effective.

Guidance for code verification activities is given in the software test specification.

General guidance for code verification activities may be provided in a generic document, such as the software verification plan (and associated procedures); specific guidance for individual tests should be provided in test specifications where necessary.

6.2.3.1 Software test specification

The software test specification is based on a general description of the software being tested. The description is supplied by the design team.

This subclause is concerned with software testing, however for many development life cycles the only software test may be the software module test (following this test the software may be tested in association with the target hardware, i.e. integration testing).

The software test specification shall include:

a) the environment in which the tests are run;

b) the test procedures;

c) acceptance criteria i.e. a detailed definition of the criteria to be fulfilled in order to accept modules and major software components on subsystem and system level;

d) error detection and corrective action procedures;

e) a list of all documents that should be produced during the code verification phase.

Non-compliance should not be raised regarding this clause providing that items a) to e) are specified at some suitable location within the documentation hierarchy.

6.2.3.2 Software test report

The software test report shall present the results of the test program described in the software test specification stating whether or not the software has achieved the performance requirements given in the software functional requirements. This document (F.M4) shall allow the complete diagnosis of design and performance deficiencies. It shall also include the resolution of all software deficiencies and test discrepancies discovered during the test.

The software test report shall include the following items both for the module and major design levels:

Providing all the above information is available in a documented format it is not necessary for it all to be contained in one report.

a) hardware configuration used for the test;

The configuration may be documented in the test specification (or some other appropriate document) providing that the configuration is verified before testing is performed.

Référence CEI

b) moyen de stockage et conditions d'accès au code final à tester;

Dans de nombreux systèmes, le moyen de stockage n'a pas d'influence sur le processus de test et aucune non-conformité majeure ne devrait être prononcée en relation avec ce paragraphe à condition que l'intégrité du logiciel en essai soit maintenue sous le contrôle de procédures de contrôle de sûreté/configuration appropriées.

c) liste des éléments en entrée du test;

Il est admis que la liste des éléments en entrée du test soit précisée dans les spécifications de test (ou tout autre document approprié).

d) liste des éléments en sortie du test;

Il est admis que la liste des éléments en sortie du test soit précisée dans les spécifications de test (ou tout autre document approprié).

e) données supplémentaires en ce qui concerne la chronologie, la séquence d'événements, etc.

Il convient de fournir ces informations le cas échéant. Il est admis que la liste des éléments en sortie du test soit précisée dans les spécifications de test (ou tout autre document approprié).

f) conformité avec les critères d'acceptation donnés dans la spécification du test;

Il peut être approprié de signaler une non-conformité par exception à condition que des preuves objectives du respect des critères d'acceptation soient disponibles pour un audit. Il convient de fournir ces preuves dans un document approprié (typiquement la spécification des tests réalisés).

g) relevé des erreurs décrivant les caractéristiques de chaque erreur et les remèdes mis en oeuvre par l'équipe de développement.

Un relevé de tous les comptes rendus d'erreurs doit être maintenu à jour. Les actions correctives qui ont été prises doivent être enregistrées de façon formelle (généralement sur la fiche de compte rendu d'erreurs remplie). Ces enregistrements doivent être présentés dans un format auditable.

7 Intégration matériel/logiciel

7.1 Plan d'intégration du système

Lors du processus de test des modules matériels et logiciels, certains aspects de conception de ces modules peuvent être mieux testés au niveau du système intégré. Mais s'il n'est pas possible de tester complètement à ce niveau, toutes les spécifications fonctionnelles de conception relatives aux modules individuels doivent être testées par d'autres moyens. Toutes les interdépendances entre la vérification des modules individuels et la vérification du système intégré doivent être documentées dans le plan d'intégration du système.

Il convient de spécifier entièrement les exigences d'interface des composants matériels et logiciels génériques. Il est préférable de tester l'implémentation complète de ces exigences avant d'intégrer les composants dans le système (ce qui peut impliquer l'utilisation de simulateurs de liaison de données, d'éléments de remplacement, de logiciels de test, etc. pour simuler la présence d'autres composants du système). Il devrait être possible de vérifier toutes les caractéristiques du logiciel au cours des essais de module mais s'il existe des caractéristiques spécifiques qu'il n'est pas possible de vérifier, il convient de les tester en utilisant un système intégré; ces éléments peuvent être identifiés par exception.

IEC reference

b) storage medium used and access requirements of the final code tested;

For many systems the storage medium is of no relevance to the testing process and no major non-compliance should be declared regarding this subclause providing that the integrity of the software under test is maintained under the control of appropriate security/configuration control procedures.

c) input test listing;

The input listing may be specified in test specifications (or some other appropriate document).

d) output test listing;

The output listing may be specified in test specifications (or some other appropriate document).

e) additional data regarding timing, sequence of events, etc.

This information should be provided if appropriate. This information may be provided in test specifications (or some other appropriate document).

f) conformance with acceptance criteria given in the test specification;

It may be appropriate to report non-conformance by exception providing that objective evidence that acceptance criteria have been met is available for audit. This evidence should be provided in an appropriate document (typically the completed test specification).

g) error incident log which describes the character of the error and the remedies taken by the design team.

A log of all error reports raised must be maintained. The corrective actions taken must be formally recorded (usually on the completed error report form). These records must be in an auditable format.

7 Hardware/software integration

7.1 System integration plan

In the process of verifying the individual hardware and software modules, certain aspects of the design of these modules may be better tested at the integrated system level. But if it is not feasible to test fully at this level, then all functional requirements of the individual module design shall be tested by other means. All interdependencies between the verification of the individual modules and the verification testing of the integrated system shall be documented in the system integration plan.

The interface requirements for the generic hardware and software components should be completely specified. It is preferable that the full implementation of these requirements is tested before components are integrated into the system (this may involve the use of datalink simulators, stubs, test software, etc., to simulate the presence of other system components). It should be possible to verify all the characteristics of software during module testing, but if there are specific features which it is not possible to verify then these should be tested using an integrated system; such items may be identified by exception.

Référence CEI

7.2 Relations entre l'intégration du système et le développement du matériel et du logiciel

Le plan d'intégration du système doit être préparé suffisamment tôt dans le processus de développement pour permettre la prise en compte des exigences liées à l'intégration, lors de la conception du matériel et du logiciel.

Il est admis que le document de conception du système général identifie toutes les exigences d'interface. Il n'est pas nécessaire qu'un plan d'intégration duplique ces informations.

7.3 Gestion de configuration du système

Le plan d'intégration du système doit établir une base de données de modules logiciels et matériels, en tant qu'outil de gestion de configuration du système. Cet outil doit permettre la gestion de versions de tous les modules matériels et logiciels utilisés dans le système.

Il est admis de définir les exigences de gestion de configuration dans un autre document approprié tel qu'un plan qualité. De même, il n'est pas nécessaire d'établir pour le plan d'intégration du système une base de données si le système doit être produit par une organisation avec un système de gestion de configuration générique établi et documenté.

Des procédures adéquates doivent être établies pour mettre en œuvre cette base de données et réaliser les tâches suivantes:

– *la procédure doit s'assurer qu'aucune insertion ou révision de module puisse être introduite si elle n'a pas été vérifiée;*

Il est admis d'entretenir une base de données contenant des modules séparés vérifiés et non vérifiés à condition qu'il soit possible de s'assurer que le système final est construit à partir de modules vérifiés.

– *la procédure doit fournir l'indice d'utilisation de chaque module du système pour permettre l'évaluation de l'incidence des révisions futures.*

Plutôt que de gérer un indice d'utilisation, il est admis de pouvoir récupérer ces informations en effectuant une consultation de la base de données ou une recherche de fichiers.

7.4 Phase d'intégration du système

Les procédures spécifiques d'intégration du matériel et du logiciel dépendent nécessairement de l'architecture du système et ne peuvent être incluses dans cette norme. Cependant, de telles procédures doivent être établies et documentées dans le plan d'intégration et doivent couvrir les activités suivantes:

– *l'acquisition de modules adéquats à la bonne version à partir de la base de données conformément aux procédures du paragraphe 7.3;*
– *l'intégration des modules matériels dans le système (par exemple position des modules, adresse mémoire, adresse module, câblage);*
– *l'édition de liens des modules logiciels et le chargement du logiciel dans le matériel;*
– *le test fonctionnel préliminaire du système intégré (voir dispositions ci-après);*
– *la documentation du processus d'intégration et de la configuration du système qui seront soumis à vérification;*
– *la mise à disposition formelle du système intégré pour permettre le test de vérification.*

Ces procédures doivent exister, mais il n'est pas nécessaire de les «établir» par le plan d'intégration du système; par exemple, des procédures existant au niveau des services/ sociétés peuvent couvrir toutes les activités ci-dessus.

IEC reference

7.2 Relationship of system integration to hardware and software development

The system integration plan shall be prepared sufficiently early in the development process to allow any integration requirements to be included in the design of the hardware and software.

It is permissible for the top level system design document to identify all interface requirements. An integration plan is not required to duplicate this information.

7.3 System configuration control

The system integration plan shall establish a library of software and hardware modules as the means for system configuration control. This library shall provide revision control for all hardware and software modules to be used in the system.

It is permissible for the configuration control requirements to be defined in some other relevant document such as a quality plan. Also, it is not necessary for the system integration plan to establish a library if the system is to be produced by an organization with an established and documented generic configuration control system.

Adequate procedures shall be established to implement this library function and shall provide for the following tasks of the library function:

– *the procedure shall ensure that no module or revision is entered into the library until it has been verified;*

It is permissible to maintain a library containing segregated verified and unverified modules providing that it is possible to ensure that the final system is built from the verified modules.

– *the procedure shall provide for an index of the use of each module in the system so that the impact of future revisions to the modules can be adequately assessed.*

Rather than maintaining an index it is permissible for such information to be retrieved by performing a database query or file search.

7.4 System integration phase

The specific procedures for the hardware/software integration necessarily depend on the nature of the system design and cannot be included in this standard. However, such procedures shall be established and documented by the integration plan, and shall cover the following activities:

– *the acquisition of the proper modules using the library and procedures of subclause 7.3;*
– *the integration of the hardware modules into the system (e.g. module position, memory address, selection, interconnection wiring);*
– *the linkage of software modules and the loading of the software into the hardware;*
– *the preliminary functional test of the integrated system function (see requirements below);*
– *the documentation of the integration process and the system configuration that will be subjected to verification testing;*
– *the formal release of the integrated system to verification testing.*

These procedures must exist, but it is not necessary for them to be “established” by the system integration plan, e.g. existing departmental/company procedures may cover all of the above activities.

Référence CEI

7.5 Vérification du système intégré

La vérification du système est le processus consistant à déterminer si les modules matériels et logiciels vérifiés ont été correctement intégrés dans le système et si le matériel et le logiciel sont compatibles et réalisent un système conformément à la spécification d'intégration.

Il est admis (et même souhaitable) de spécifier toute exigence d'intégration, c'est-à-dire toute exigence du système intégré, parmi les exigences fonctionnelles du système. Il n'est pas nécessaire de les dupliquer dans un ensemble distinct d'exigences d'intégration.

Le système doit être aussi complet que possible pour ce test.

Les jeux d'essais choisis pour la vérification du système doivent activer toutes les interfaces des modules ainsi que les fonctions de base des modules eux-mêmes. La simulation d'une partie du système ou de ses interfaces doit être justifiée dans le plan d'intégration du système.

Il n'est pas possible de concevoir le test d'intégration du système pour tester les interfaces de modules logiciels. Le but de ce test est de tester toutes les interfaces de composants (les composants peuvent contenir plusieurs modules logiciels) dans une configuration de système se rapprochant étroitement du système final. Cette approche est cohérente avec une approche de test «ascendante» et cette approche est raisonnable à condition que les interfaces des modules aient été testées au cours d'une phase de test antérieure.

7.7 Vérification du système intégré, compte rendu de test

Les résultats de la vérification du système intégré doivent être inclus dans un compte rendu de test (F.M5). Celui-ci doit identifier le matériel et le logiciel utilisés, l'équipement d'essai utilisé et son réglage, la simulation de parties du système ou des interfaces, les résultat incorrects rencontrés en cours de vérification avec les actions correctives entreprises conformément au paragraphe 7.6. Les résultats d'essai doivent être consignés sous une forme permettant l'audit par des personnes qui ne sont pas directement impliquées dans le programme d'essai.

La correction de toutes les erreurs consignées et les résultats de l'évaluation qui en découlent doivent être documentés avec suffisamment de détails et sous une forme permettant l'audit par des personnes qui ne sont pas directement impliquées dans le développement du système et le programme de vérification.

Il est admis de spécifier les configurations matérielles et logicielles dans la documentation d'essai et de les faire confirmer par les testeurs du système. Le matériel d'essai peut être obtenu aupres d'un magasin central de matériel étalonné; il convient que ce type de matériel porte des étiquettes identifiant l'étalonnage et qu'il puisse être audité. Il est admis que les procédures des sociétés garantissent exclusivement l'utilisation d'un matériel correctement étalonné dans le cadre des tests.

Des écarts par rapport aux recommandations détaillées de la Norme seraient généralement admis à condition de disposer d'informations suffisamment documentées afin de permettre de répéter les essais en utilisant une configuration identique des composants logiciels et matériels.

8 Validation du système programmé

Des essais doivent être réalisés pour valider le matériel et le logiciel en tant que système par rapport aux spécifications du système de sûreté que doit satisfaire le système programmé. La validation du système programmé doit être conduite en accord avec un plan formalisé de validation (F.M5). Ce plan doit expliciter la validation dans les conditions statiques et dynamiques.

IEC reference

7.5 Integrated system verification

The system verification is the process of determining whether or not the verified hardware and software modules have been properly integrated into the system and that the hardware and software are compatible and perform as a system as required by the integration requirements.

It should be acceptable (indeed desirable) for any integration requirements, i.e. requirements for the integrated system, to be specified in the system functional requirements. It should not be necessary for these to be duplicated within a separate set of integration requirements.

The system shall be as complete as is practical for this testing.

The test cases selected for system verification shall exercise all module interfaces as well as the basic operation of the modules themselves. Justification shall be provided in the system integration plan for the simulation of any part of the system or its interfaces.

It is not possible to design the system integration test to exercise the software module interfaces. The purpose of this test is to test all the component (components may contain many software modules) interfaces in a system configuration which closely resembles the final system. This approach is consistent with a “bottom up” testing approach and this approach is reasonable providing the module interfaces have been tested during a previous testing phase.

7.7 Integrated system verification test report

The results of the integrated system verification shall be documented in a test report (F.M5). This report shall identify the hardware and software used, the test equipment and its calibration, the simulation of system or interface components, and any test results discrepancy found along with the corrective actions taken according to subclause 7.6. The test results shall be retained in a form that is auditable by persons not directly engaged in the test program.

The resolution of all reported errors and the results of the subsequent evaluation shall be documented in sufficient detail and in a manner that is auditable by persons not directly engaged in the system development and verification program.

The hardware and software configurations may be specified in the test documentation and confirmed by the system testers. Test equipment may be obtained from a central store of calibrated equipment, all such equipment should have labels identifying calibration status and be available for audit. There may be company procedures which ensure that only correctly calibrated equipment is used to conduct tests.

Deviations from the detailed recommendations of the Standard would be generally acceptable providing that sufficient documented information is available to enable any test to be repeated using an identical configuration of software and hardware components.

8 Computer system validation

Testing shall be performed to validate the hardware and software as a system in accordance with the safety systems requirements to be satisfied by the computer system. The computer system validation shall be conducted in accordance with a formal validation plan (F.M5). The plan shall identify the validation for static and dynamic cases.

Référence CEI

La validation du système programmé est généralement un test du logiciel d'application réalisé sur le matériel cible; la configuration réelle du noyau système peut ne pas utiliser toutes les fonctionnalités de ce noyau; par exemple, un logiciel peut être fourni pour réaliser une conversion de racine carrée et être intégré à un module standard susceptible de ne pas être utilisé. Cette approche est généralement admise à condition que l'ensemble de la fonctionnalité de tous les composants du noyau ait été vérifiée au cours d'une phase de test antérieure.

Le système programmé doit être sollicité par la simulation statique et dynamique des signaux d'entrée présents durant le fonctionnement normal, ainsi que lors des incidents d'exploitation et des conditions d'accident entraînant une action du système programmé. Il est recommandé que chaque fonction de sécurité du réacteur soit contrôlée par des essais représentatifs de chaque variable intervenant dans l'arrêt d'urgence ou la protection, prise isolément et en combinaison.

Des essais ne sont pas nécessaires pour valider les composants qui ne sont pas utilisés par le système particulier en cours d'essai. Il convient de pouvoir identifier, dans le rapport de tests, quels composants n'ont pas été testés.

Il est recommandé que les essais:

– *couvrent la plage de variation des signaux et les gammes des variables élaborées ou calculées d'un manière pleinement représentative;*

Il n'est pas nécessaire de tester les paramètres calculés de façon interne au cours des tests de validation du système programmé à condition qu'une phase de test antérieure ait vérifié le fonctionnement correct des calculs internes.

– *couvrent les circuits de vote et les autres circuits logiques le plus complètement possible;*
– *soient effectués pour tous les signaux d'arrêt d'urgence ou de protection dans la configuration finale du système;*
– *donnent l'assurance que la précision et les temps de réponse sont respectés et qu'une action correcte est effectuée dans le cas d'une panne quelconque de l'équipement ou d'une combinaison de pannes.*

Il convient de reconnaître qu'il n'est pas faisable de tester toutes les combinaisons possibles de pannes de matériel pour des systèmes importants. Ces tests ne sont pas considérés comme étant nécessaires pour des systèmes modulaires où les effets de la défaillance de composants particuliers peuvent être déterminés. Il convient de tester la réponse du système à un échantillon représentatif de pannes de composants.

De plus, les valeurs des signaux d'entrée requis, les signaux de sortie attendus et les critères d'acceptation doivent être spécifiés dans le plan de validation. Celui-ci doit être mis en oeuvre et les résultats de la validation évalués par des personnes qui n'ont pas participé à la phase de développement décrite dans l'article 5.

Il est admis de documenter les signaux de sortie attendus dans les spécifications d'essai de validation. Il est admis que les spécifications d'essai de validation soient produites par des équipes qui comprennent le personnel de développement à condition que les spécifications soient soumises à une revue effectuée par un personnel non directement impliqué dans la conception et le codage du logiciel. Il convient de faire intervenir des personnes non impliquées dans la conception et le codage du logiciel au niveau des tests, soit pour les réaliser, soit pour assister à leur déroulement.

Les équipements utilisés pour la validation doivent être étalonnés. Les outils matériels et logiciels utilisés pour cette validation ne nécessitent pas de vérification spéciale. Il convient cependant de pouvoir montrer qu'ils remplissent leur rôle.

IEC reference

Computer system validation is usually a test of the application software running in target hardware; the actual configuration of the kernel system may not use all the functionality of the kernel, e.g. there may be software to perform square root conversion provided as part of a standard module which may never be run. This approach would be generally acceptable providing that the full functionality of all kernel components has been verified at a previous testing phase.

The computer system shall be exercised by static and dynamic simulation of input signals present during normal operation, anticipated operational occurrences and accident conditions requiring computer system action. Each reactor safety function should be confirmed by representative tests of each trip or protection parameter singly and in combination.

Tests are not required to validate components which are not used in the particular system under test. It should be possible to identify which components were not tested from the test records.

It is recommended that the tests:

– *cover all signal ranges, and the ranges of computed or calculated parameters in a fully representative manner;*

It is not necessary to test internally calculated parameters during computer system validation testing providing that a previous testing phase has verified the correct operation of internal calculations.

– *cover the voting and other logic combinations comprehensively;*
– *be made for all trip or protective signals in the final assembly configuration;*
– *ensure that accuracy and response times are confirmed, and that correct action is taken for any equipment failure or failure combination.*

It should be acknowledged that it will not be feasible to test all possible combinations of equipment failures for large systems. Such tests are not considered necessary for modular systems where the effects of the failure of particular components can be determined. The response of the system to a representative sample of component failures should be tested.

In addition, the required input signals and their values, the anticipated output signals and the acceptance criteria shall be stated in the validation plan. The computer system validation plan shall be developed and the results of the validation evaluated by individuals who did not participate in the development phase of clause 5.

The anticipated output signals may be documented in the validation test specifications. The validation test specifications may be developed by teams which include development staff providing the specifications are subject to review by personnel not directly involved in the design and coding of the software. Testing should be performed, or witnessed, by individuals who did not design and code the software.

Equipment used for validation shall be calibrated. Hardware and software tools used for this validation need no special verification. They should, however, be shown to be suited to their purpose.

Référence CEI

9 Maintenance et modification

9.1.1 La modification peut être demandée à la suite d'un compte rendu d'anomalie consécutif à des essais. La modification due à ces causes peut occasionner un changement dans l'analyse fonctionnelle du logiciel. La modification peut aussi être demandée après la livraison de l'équipement, il s'agit alors de maintenance de logiciel.

Il est admis de considérer qu'à partir de la date de prise en charge du système par les acheteurs, les changements apportés au système constituent une maintenance plutôt qu'une modification. La prise en charge peut survenir quelque temps après la livraison du système sur site du fait de l'ampleur des tests généralement nécessaires.

9.1.2 La demande de modification doit être évaluée de manière indépendante, il peut en résulter:

– *le rejet de la demande; dans ce cas, elle est retournée avec des commentaires;*
– *l'acceptation d'une demande de modification mineure et sa mise en oeuvre immédiate si elle intervient lors du développement initial du logiciel et cela après analyse;*
– *la demande d'une analyse détaillée des incidences conduisant à l'élaboration d'un compte rendu d'analyse des incidences sur le logiciel. Ce compte rendu doit être rédigé par un spécialiste du logiciel du système;*
– *l'acceptation de la demande.*

Il convient de déterminer le niveau de revue indépendante en fonction de l'ampleur de la modification; par exemple, il convient qu'une modification affectant la sûreté d'une centrale soit examinée par les personnes responsables de la sûreté, et qu'une modification affectant une donnée mineure, telle que le domaine de validité pour un signal d'entrée, soit seulement soumise à l'accord de l'ingénieur responsable du système. Une approche graduée pour la revue des demandes de modification est généralement acceptable à condition que les principes ci-dessus soient respectés, c'est-à-dire que chaque modification reçoive un niveau de revue/autorisation adéquat (et soit cohérent avec 9.2.2).

NOTE – Il n'est pas souhaitable de faire examiner les modifications par un personnel n'apportant rien au processus de revue, par exemple de faire examiner une plage de variation des signaux de pression vapeur à des spécialistes en physique nucléaire. Si les revues sont excessivement développées, le processus de revue peut se transformer en une activité principalement administrative pouvant avoir un effet globalement néfaste au niveau de la sûreté de l'organisation, en détournant le personnel d'un travail important – les sujets de sûreté critiques peuvent être négligés en raison de la surcharge de travail administratif créée.

9.1.3 Les points suivants doivent être examinés pour l'évaluation de la demande de modification:

– *la faisabilité;*

Le respect de ce paragraphe n'est pas considéré comme un sujet de sûreté – les modifications techniquement infaisables ne dépassant pas la phase de test hors ligne, aucune non-conformité majeure ne devrait être prononcée en relation avec ce paragraphe.

– *les conséquences sur le matériel (par exemple extension mémoire) et/ou sur d'autres équipements (par exemple équipements de tests). Dans ce cas, la demande correspondante de modification doit être explicitée au niveau équipement;*
– *les conséquences sur le logiciel, comprenant une liste des modules affectés;*

Le respect de ce paragraphe n'est pas considéré comme un sujet de sûreté; et aucune non-conformité majeure ne devrait être prononcée en relation avec ce paragraphe à condition que les modules affectés soient identifiés au cours d'une phase ultérieure d'analyse détaillée des conséquences.

IEC reference

9 Maintenance and modification

9.1.1 The modification may be requested because of an anomaly report as a result of tests. Modification for these reasons may cause a change in the software requirements document. The modification may also be requested after delivery; it is then considered to be maintenance.

The date of system takeover by the purchasers may be considered the time from which changes to the system are classified as maintenance rather than modification. Takeover may occur some time after delivery of the system to site due to the extensive site testing generally required.

9.1.2 The modification request shall be evaluated independently, the result being either:

– *to reject the request; in this case it is sent back with comments;*
– *to approve a request of minor importance and impact immediately if it is made during initial development after analysis;*
– *to require a detailed analysis resulting in a software modification analysis report. This report shall be written by software personnel knowledgeable in the system software;*
– *to accept the request.*

The level of independent review required should depend upon the extent of change, e.g. a change affecting the safety case for a station should be reviewed by those responsible for the safety case, a change affecting a minor data item, such as the valid range of an input signal, should only require approval by the responsible system engineer. A graded approach to change request review is generally acceptable providing that the above principles are followed, i.e. each change receives an appropriate level of review/authorization (and is consistent with 9.2.2).

NOTE – It is not desirable for changes to be subject to review by personnel who have no added value to give to the review process, e.g. to have a steam pressure signal range change reviewed by reactor physics specialists. If reviews are overly extensive the review process can become a mainly administrative exercise which may have a detrimental overall effect upon the organization's safety culture, i.e. detract skilled staff from important work – the critical safety issues may be overlooked due to the excessive administrative work load created.

9.1.3 The following items shall be examined in the evaluation of the modification request:

– *technical feasibility;*

Observance of this subclause is not considered to be a safety issue – technically unfeasible changes will not pass the off-line testing phase and for this reason no major non-compliance should be raised against this clause.

– *impact upon hardware (e.g. memory extension) or upon other equipment (e.g. test systems) in which case the request for modification addressing this impact area must be documented for the equipment;*
– *impact upon software including a list of affected modules;*

Observance of this subclause is not considered to be a safety issue; and no major non-compliance should be declared regarding this subclause providing the affected modules are identified during a subsequent detailed impact analysis phase.