

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC
880**

Première édition
First edition
1986

**Logiciel pour les calculateurs utilisés dans
les systèmes de sûreté des centrales nucléaires**

**Software for computers in the safety systems
of nuclear power stations**



Numéro de référence
Reference number
CEI/IEC 880: 1986

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 auprès du Bureau Central 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:

- **Bulletin de la CEI**
- **Annuaire de la CEI**
Publié annuellement
- **Catalogue des publications de la CEI**
Publié annuellement et mis à jour régulièrement

Terminologie

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 50: *Vocabulaire Electrotechnique International* (VEI), qui se présente sous forme de chapitres séparés traitant chacun d'un sujet défini. Des détails complets sur le VEI peuvent être obtenus sur demande. Voir également le dictionnaire multilingue de la CEI.

Les termes et définitions figurant dans la présente publication ont été soit tirés du VEI, soit spécifiquement approuvés aux fins de cette publication.

Symboles graphiques et littéraux

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera:

- la CEI 27: *Symboles littéraux à utiliser en électro-technique;*
- la CEI 417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles;*
- la CEI 617: *Symboles graphiques pour schémas;*

et pour les appareils électromédicaux,

- la CEI 878: *Symboles graphiques pour équipements électriques en pratique médicale.*

Les symboles et signes contenus dans la présente publication ont été soit tirés de la CEI 27, de la CEI 417, de la CEI 617 et/ou de la CEI 878, soit spécifiquement approuvés aux fins de cette publication.

Publications de la CEI établies par le même comité d'études

L'attention du lecteur est attirée sur les listes figurant à la fin de cette publication, qui énumèrent les publications de la CEI préparées par le comité d'études qui a établi la présente publication.

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 from the IEC Central Office.

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 Bulletin**
- **IEC Yearbook**
Published yearly
- **Catalogue of IEC publications**
Published yearly with regular updates

Terminology

For general terminology, readers are referred to IEC 50: *International Electrotechnical Vocabulary* (IEV), which is issued in the form of separate chapters each dealing with a specific field. Full details of the IEV will be supplied on request. See also the IEC Multilingual Dictionary.

The terms and definitions contained in the present publication have either been taken from the IEV or have been specifically approved for the purpose of this publication.

Graphical and letter symbols

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications:

- IEC 27: *Letter symbols to be used in electrical technology;*
- IEC 417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets;*
- IEC 617: *Graphical symbols for diagrams;*

and for medical electrical equipment,

- IEC 878: *Graphical symbols for electromedical equipment in medical practice.*

The symbols and signs contained in the present publication have either been taken from IEC 27, IEC 417, IEC 617 and/or IEC 878, or have been specifically approved for the purpose of this publication.

IEC publications prepared by the same technical committee

The attention of readers is drawn to the end pages of this publication which list the IEC publications issued by the technical committee which has prepared the present publication.

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC
880**

Première édition
First edition
1986

**Logiciel pour les calculateurs utilisés dans
les systèmes de sûreté des centrales nucléaires**

**Software for computers in the safety systems
of nuclear power stations**

© CEI 1986 Droits de reproduction réservés — Copyright - all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher

Bureau central de la Commission Electrotechnique Internationale 3, rue de Varembe Genève Suisse



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE **XB**

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

SOMMAIRE

	Pages
PRÉAMBULE	4
PRÉFACE	4
INTRODUCTION	6
Articles	
1. Domaine d'application et objet	6
2. Termes et définitions	8
3. Structure du projet	12
4. Spécifications du logiciel	12
5. Développement (conception et codage) du logiciel des systèmes de sûreté	16
6. Vérification	22
7. Intégration matériel/logiciel	26
8. Validation du système programmé	32
9. Maintenance et modification	34
10. Exploitation	38
ANNEXE A — Synoptique du cycle de développement du système et détails des spécifications du logiciel	42
ANNEXE B — Recommandations détaillées pour le développement (conception et programmation) des logiciels de sûreté	56
ANNEXE C — Grandes lignes des spécifications du logiciel	94
ANNEXE D — Langage, traducteur, éditeur de liens, etc.	98
ANNEXE E — Test du logiciel	102
ANNEXE F — Liste des documents nécessaires	116
INDEX ALPHABÉTIQUE	124

CONTENTS

	Page
FOREWORD	5
PREFACE	5
INTRODUCTION	7
Clause	
1. Scope and object	7
2. Terms and definitions	9
3. Project structure	13
4. Software requirements	13
5. Development (design and coding) of safety system software	17
6. Verification	23
7. Hardware/software integration	27
8. Computer system validation	33
9. Maintenance and modification	35
10. Operation	39
APPENDIX A — System development life cycle and details of software requirements	43
APPENDIX B — Detailed recommendations for the development (design and coding) of safety related software	57
APPENDIX C — Outline for the software performance specification	95
APPENDIX D — Language, translator, linkage editor, etc.	99
APPENDIX E — Software testing	103
APPENDIX F — List of documents needed	117
ALPHABETICAL INDEX	129

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

LOGICIEL POUR LES CALCULATEURS UTILISÉS DANS LES SYSTÈMES DE SÛRETÉ DES CENTRALES NUCLÉAIRES

PRÉAMBULE

- 1) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par des Comités d'Etudes où sont représentés tous les Comités nationaux s'intéressant à ces questions, expriment dans la plus grande mesure possible un accord international sur les sujets examinés.
- 2) Ces décisions constituent des recommandations internationales et sont agréées comme telles par les Comités nationaux.
- 3) Dans le but d'encourager l'unification internationale, la CEI exprime le vœu que tous les Comités nationaux adoptent dans leurs règles nationales le texte de la recommandation de la CEI, dans la mesure où les conditions nationales le permettent. Toute divergence entre la recommandation de la CEI et la règle nationale correspondante doit, dans la mesure du possible, être indiquée en termes clairs dans cette dernière.

PRÉFACE

La présente norme a été établie par le Sous-Comité 45A: Instrumentation des réacteurs, du Comité d'Etudes n° 45 de la CEI: Instrumentation nucléaire.

Le texte de cette norme est issu des documents suivants:

Regle des Six Mois	Rapport de vote
45A(BC)88-I/II/III	45A(BC)90

Pour de plus amples renseignements, consulter le rapport de vote mentionné dans le tableau ci-dessus.

Les publications suivantes de la CEI sont citées dans la présente norme:

Publications n° 557 (1982): Terminologie CEI sur les réacteurs nucléaires.

639 (1979): Réacteurs nucléaires. Utilisation du système de protection à d'autres fins que la sécurité.

643 (1979): Application des calculateurs numériques à l'instrumentation et à la conduite des réacteurs nucléaires.

671 (1980): Essais périodiques et surveillance du système de protection des réacteurs nucléaires.

Autres publications citées:

Norme ISO 2382/1 (1984): Traitement des données — Vocabulaire — Partie 01: Termes fondamentaux.

Guide de l'AIEA 50-SG-D3 (1980): Guide de sûreté.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**SOFTWARE FOR COMPUTERS IN THE SAFETY SYSTEMS
OF NUCLEAR POWER STATIONS**

FOREWORD

- 1) The formal decisions or agreements of the IEC on technical matters, prepared by Technical Committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.
- 2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.
- 3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules in so far as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rules should, as far as possible, be clearly indicated in the latter.

PREFACE

This standard has been prepared by Sub-Committee 45A: Reactor Instrumentations, of IEC Technical Committee No. 45: Nuclear Instrumentation.

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

Six Months' Rule	Report on Voting
45A(CO)88-I/II/III	45A(CO)90

Further information can be found in the Report on Voting indicated in the table above.

The following IEC publications are quoted in this standard:

- Publications Nos. 557 (1982): IEC Terminology in the Nuclear Reactor Field.
639 (1979): Nuclear Reactor. Use of the Protection System for Non-safety Purposes.
643 (1979): Application of Digital Computers to Nuclear Reactor Instrumentation and Control.
671 (1980): Periodic Tests and Monitoring of the Protection System of Nuclear Reactors.

Other publications quoted:

- ISO standard 2382/1 (1984): Data processing — Vocabulary — Part 01: Fundamental terms.
IAEA Guide 50-SG-D3 (1980): Safety guide.

LOGICIEL POUR LES CALCULATEURS UTILISÉS DANS LES SYSTÈMES DE SÛRETÉ DES CENTRALES NUCLÉAIRES

INTRODUCTION

Les principes de base pour la conception de l'instrumentation nucléaire appliqués en particulier aux systèmes de sûreté des centrales nucléaires ont été traités dans les normes existantes, se référant aux systèmes câblés tel le «Guide de sûreté 50-SG-D3» de l'AIEA.

La présente norme a été développée pour interpréter ces principes lors de l'utilisation de systèmes programmés — systèmes multiprocesseurs distribués aussi bien que gros processeurs centraux — dans les systèmes de sûreté des centrales nucléaires. Elle traite des principes et prescriptions relatives au logiciel du système et il convient qu'elle soit employée en association avec les normes appropriées traitant du matériel et de l'intégration du système.

Il est important de noter que cette norme n'introduit pas d'exigences fonctionnelles supplémentaires pour les systèmes de sûreté.

Les aspects qui ont été spécialement envisagés, vu le caractère particulier des systèmes programmés et de leur logiciel, sont:

- a) les critères auxquels doit répondre le matériel pour autant qu'ils affectent le logiciel, en tenant compte du haut degré d'interdépendance entre le matériel et le logiciel;
- b) une approche générale du développement du logiciel pour assurer la production d'un logiciel de grande fiabilité;
- c) une approche générale de la vérification du logiciel et de la validation du système programmé;
- d) les procédures relatives à la maintenance du logiciel, la modification, la gestion de la configuration.

1. Domaine d'application et objet

Cette norme est applicable au logiciel de haute fiabilité exigé pour les équipements programmés devant être utilisés dans les systèmes de sûreté des centrales nucléaires pour les fonctions liées à la sûreté — fonctions de classe 1, selon la Publication 643 de la CEI: Application des calculateurs numériques à l'instrumentation et à la conduite des réacteurs nucléaires. Cela inclut le système de commande, les systèmes auxiliaires des systèmes de sauvegarde et les systèmes de protection.

Pour les fonctions qui ne sont pas de sûreté, les principes de la Publication 639 de la CEI: Réacteurs nucléaires — Utilisation du système de protection à d'autres fins que la sûreté, sont applicables dans le cas d'utilisation de systèmes programmés.

Cette norme fournit un ensemble de prescriptions pour chaque étape de l'élaboration du logiciel, comprenant la conception, le développement, la qualification et la mise en œuvre, ainsi que la documentation accompagnant chaque étape de l'élaboration du logiciel dans le but d'obtenir un logiciel de haute fiabilité.

Les principes appliqués dans la mise en œuvre de ces prescriptions sont les suivants:

- méthodes les plus avancées;
- conception descendante;
- modularité;
- vérification de chaque phase;
- documentation claire;
- documents pouvant être soumis à audit;
- contrôle de validation.

SOFTWARE FOR COMPUTERS IN THE SAFETY SYSTEMS OF NUCLEAR POWER STATIONS

INTRODUCTION

The basic principles for the design of nuclear instrumentation as specifically applied to the safety systems of nuclear power plants have been interpreted in existing standards with reference to hard-wired systems as the "Safety Guide 50-SG-D3" of the IAEA.

This standard has been developed to interpret these principles for the utilization of digital systems — multiprocessor distributed systems as well as larger scale central processor systems — in the safety systems of nuclear power plants. It discusses the software system principles and requirements and should be read in association with appropriate standards on computer hardware and system integration.

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

Aspects for which special recommendations have been produced, due to the unique nature of computer systems and their software are:

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

1. Scope and object

This standard is applicable to highly reliable software required for computers to be used in the safety systems of nuclear power plants for safety functions — Class 1 functions according to IEC Publication 643: Application of Digital Computers to Nuclear Reactor Instrumentation and Control. This includes the safety actuation systems, the safety system support features and the protection systems.

For the utilization of computer systems for non-safety functions the principles of IEC Publication 639: Nuclear Reactors — Use of the Protection System for Non-safety Purposes, are applicable.

This standard provides requirements for each stage of software generation, including design, development, qualification and operation as well as the documentation for each stage of the software generation for the purpose of achieving highly reliable software.

The principles applied in developing these requirements include:

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

Des directives et informations complémentaires sur la manière de satisfaire aux prescriptions de la partie principale de cette norme sont données dans les annexes A à F. Les références à ces annexes sont données entre parenthèses.

Si des pratiques différentes de celles figurant dans les annexes sont utilisées, elles doivent être documentées et pouvoir être vérifiées par rapport aux exigences de la partie principale de cette norme.

2. Termes et définitions

Les termes utilisés sont conformes à la Norme ISO 2382 (jusqu'à 1984), sauf en ce qui concerne les termes suivants:

2.1 *Programme d'application*

Programme d'ordinateur accomplissant une tâche liée au processus à commander plutôt qu'au fonctionnement de l'ordinateur lui-même.

2.2 *Compactage du code*

Réduction réfléchie de la taille mémoire nécessitée par un programme obtenue par élimination des instructions redondantes ou sans rapport avec ledit programme.

2.3 *Ordinateur ou calculateur (Réf. Norme ISO 2382/1)*

Unité fonctionnelle programmable se composant d'une ou de plusieurs unités centrales associées et de périphériques, qui est commandée par des programmes rangés en mémoire interne et qui est capable d'effectuer des calculs importants, comportant de nombreuses opérations arithmétiques ou des opérations logiques, sans intervention humaine pendant l'exécution.

Note. — Un ordinateur peut se composer d'une seule ou de plusieurs unités interconnectées.

2.4 *Programme d'ordinateur*

Ensemble ordonné d'instructions et de données qui spécifient des opérations sous une forme acceptable pour exécution par un système programmé.

2.5 *Système programmé*

Ensemble fonctionnel constitué d'une ou de plusieurs unités fonctionnelles et du logiciel associé qui utilise des éléments de stockage communs pour tout ou partie d'un programme ainsi que pour tout ou partie des données nécessaires à l'exécution du programme.

2.6 *Données*

Représentation des faits, concepts ou instructions selon une manière formalisée convenable pour la communication, l'interprétation ou le traitement par un ordinateur.

2.7 *Défense en profondeur*

Mise en œuvre, vis-à-vis d'un seuil, de plusieurs barrières successives se recouvrant de telle manière que ce seuil ne puisse être franchi que dans le cas où toutes les barrières sont défaillantes.

2.8 *Diversité*

Existence de différents moyens de réaliser une fonction requise (par exemple autres principes physiques, autres manières de résoudre la même tâche).

2.9 *Tolérances aux fautes*

Capacité intrinsèque d'un système de fonctionner correctement de manière continue en présence d'un nombre limité de fautes concernant le matériel ou le logiciel.

Additional guidance and information on how to comply with the requirements of the main part of this standard is given in Appendices A to F. References to those appendices are given in brackets.

If practices differing from those of the appendices are used, they shall be documented and auditable according to the requirements of the main part of this standard.

2. Terms and definitions

Terms are used as defined in ISO Standard 2382 (up to 1984), except as given below.

2.1 *Application program*

A computer program that performs a task related to the process being controlled rather than to the functioning of the computer itself.

2.2 *Code compaction*

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

2.3 *Computer (Ref. ISO Standard 2382/1)*

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

Note. — A computer may be a stand-alone unit or may consist of several interconnected units.

2.4 *Computer program*

A set of ordered instructions and data that specify operations in a form suitable for execution by a digital computer.

2.5 *Computer system*

A functional unit, consisting of one or more computers and associated software, that uses common storage for all or part of a program and also for all or part of the data necessary for the execution of the program.

2.6 *Data*

A representation of facts, concepts or instructions in a formalized manner suitable for communication, interpretation or processing by a computer.

2.7 *Defence in depth*

Provision of several overlapping subsequent limiting barriers with respect to one threshold, such that the threshold can only be surpassed, if all barriers have failed.

2.8 *Diversity*

Existence of different means of performing a required function (e.g. other physical principles, other ways of solving the same task).

2.9 *Fault tolerance*

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

2.10 *Marche dégradée*

Réduction par étapes des fonctions du système en réponse à une panne détectée alors que les fonctions essentielles sont maintenues.

2.11 *Initialiser*

Forcer les compteurs, clés, adresses ou contenus des organes de stockage à des valeurs prédéterminées au début ou en certains points prescrits du fonctionnement d'un programme.

2.12 *Tests d'intégration*

Tests effectués lors du processus d'intégration du matériel et du logiciel avant la phase de validation du système de traitement, pour vérifier la compatibilité du logiciel et du matériel.

2.13 *Procédure*

Partie de programme d'ordinateur qui est désignée par un nom et qui réalise une tâche spécifique.

2.14 *Redondance*

Présence d'éléments ou systèmes (identiques ou différents) de rechange, de sorte que l'un d'entre eux puisse remplir la fonction requise quel que soit l'état de fonctionnement ou malgré la défaillance d'un autre (Réf. AIEA 50-SG-D8).

2.15 *Critère de défaillance unique (Publication 557 de la CEI: Terminologie CEI sur les réacteurs nucléaires)*

Critère appliqué à un système tel que celui-ci soit capable de remplir sa propre tâche de sécurité lorsqu'il est soumis à un seul défaut.

2.16 *Logiciel*

Programmes, procédures, règles et tout document associé, liés à la mise en œuvre du système programmé.

2.17 *Cycle de vie du logiciel*

Période de temps qui débute lors de la conception d'un produit logiciel et qui s'achève lorsque le produit n'est plus utilisable. Le cycle de vie du logiciel comprend une phase de spécification, une phase de conception, une phase de réalisation, une phase de test, une phase d'installation et de validation et une phase d'exploitation et de maintenance.

2.18 *Modification du logiciel*

Changement de documents déjà approuvés conduisant à un changement du code exécutable ou de ses données.

2.19 *Modularité du logiciel*

Caractéristique du logiciel qui présente une structure en unités de programme fortement indépendantes les unes des autres, autonomes et identifiables pour la compilation, le test et l'association avec d'autres unités.

2.20 *Validation*

Test et évaluation du système de traitement intégré (matériel et logiciel) pour s'assurer de sa conformité aux spécifications fonctionnelles, ainsi qu'aux spécifications de performances et d'interfaces.

2.21 *Vérification*

Processus consistant à déterminer lors de chaque phase du développement du système de traitement programmé si le produit élaboré répond aux spécifications issues de la phase précédente.

2.10 Graceful degradation

Stepwise reduction of system functions in response to detected failures while essential functions are maintained.

2.11 Initialize

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

2.12 Integration tests

Tests performed during the hardware/software integration process prior to computer system validation to verify compatibility of the software and the computer system hardware.

2.13 Procedure

A portion of a computer program which is named and which performs a specific task.

2.14 Redundancy

Provision of alternative (identical or diverse) elements or systems so that any one can perform the required function regardless of the state of operation or failure of any other (Ref. IAEA 50-SG-D8).

2.15 Single failure criterion (IEC Publication 557: IEC Terminology in the Nuclear Reactor Field)

A criterion applied to a system such that it is capable of performing its safety task in the presence of any single failure.

2.16 Software

Programs, procedures, rules and any associated documentation pertaining to the operation of a computer system.

2.17 Software life cycle

The period of time that starts when a software product is conceived and ends when the product is no longer available for use. The software life cycle typically includes a requirements phase, design phase, implementation phase, test phase, installation and check-out phase, operation and maintenance phase.

2.18 Software modification

Changes of already agreed documents leading to a change to the executable code or its data.

2.19 Software modularity

The software attribute that provides a structure of highly independent computer program units that are discrete and identifiable with respect to translating, testing and combining with other units.

2.20 Validation

The test and evaluation of the integrated computer system (hardware and software) to ensure compliance with the functional, performance and interface requirements.

2.21 Verification

The process of determining whether or not the product of each phase of the digital computer system development process fulfils all the requirements imposed by the previous phase.

3. Structure du projet

En règle générale, un projet doit être découpé en phases. Chaque phase constitue une entité mais est interdépendante des autres phases, qui elles-mêmes en dépendent. Ces phases sont identifiables de manière informelle par les activités spécifiques qui s'y rapportent.

Pour les applications liées à la sûreté, ces phases doivent être formalisées, et aucune des phases identifiées ne doit être omise.

3.1 Généralités

Les facteurs généraux suivants déterminent les activités dans l'exécution d'un projet:

- 3.1.1 L'ensemble du cycle de vie du logiciel doit être considéré.
- 3.1.2 Chaque phase du cycle de vie du logiciel doit être divisée en tâches élémentaires dont l'activité doit être bien définie.
- 3.1.3 Chaque produit doit être systématiquement contrôlé après chaque phase (B4.g).
- 3.1.4 Les tâches d'assurance qualité doivent en règle générale se dérouler en parallèle avec les autres tâches du cycle de vie du logiciel.
- 3.1.5 Chaque phase doit inclure l'élaboration des documents appropriés (F3).
- 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).
- 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.
- 3.1.8 Une liste des documents requis au minimum durant la vie du logiciel est donnée dans l'annexe F.

3.2 Plan d'assurance qualité pour le logiciel

Un plan d'assurance qualité doit exister ou être établi au début du projet soit en tant que chapitre dans les spécifications du système programmé (voir article 4), soit en tant que document d'accompagnement.

Bien que l'ensemble de cette norme et de ses annexes puisse être considéré comme un plan d'assurance qualité pour le logiciel relatif aux systèmes de sûreté, des plans de qualité spéciaux peuvent être adoptés pour les phases liées à des produits individuels pour des composants logiciels particuliers selon les normes nationales ou les règlements propres aux sociétés.

Il convient que ces plans traitent de toutes les procédures d'assurance qualité requises durant toutes les phases du cycle de vie du logiciel.

4. Spécifications du logiciel

Les spécifications du logiciel (F.M2) sont issues des spécifications des systèmes de sûreté et constituent une partie des spécifications du système programmé. Les spécifications du système programmé présentent une description à la fois du matériel et du logiciel et établissent les objectifs et les fonctions du système programmé.

Les exigences relatives au logiciel décrivent le produit, et non son développement. Elles doivent décrire ce qui doit être fait et non la manière de le faire. Une approche satisfaisante pour la définition du contenu et l'établissement des spécifications du logiciel, qui est en accord avec le présent article, est donnée dans l'annexe A.

Bien que le but principal de cette analyse fonctionnelle du logiciel soit de constituer la base de développement du logiciel, les aspects liés aux organismes de sûreté ne doivent pas être

3. Project structure

Any project will normally be divided up into a number of phases. Each phase is to some extent self-contained but will depend on other phases and will, in its turn, be depended on by others. These phases are informally recognizable by the specific activities pertinent to them.

For safety related applications, these phases shall be formalized and none of the identified phases shall be omitted.

3.1 General

The following general factors determine the activities in implementing a project:

- 3.1.1 The whole software life cycle shall be considered.
- 3.1.2 Each phase of the software life cycle shall be divided into elementary tasks with a well-defined activity for each of them.
- 3.1.3 Every product shall be systematically checked after each phase (B4.g).
- 3.1.4 The tasks of quality assurance should be run generally in parallel with the other tasks of the life cycle.
- 3.1.5 Each phase shall include generation of the appropriate documents (F3).
- 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).
- 3.1.7 Every verification step or critical review shall result in a report on the analysis performed, the conclusions reached and the resolutions agreed. This report shall be included in the documentation.
- 3.1.8 A list of documents required as a minimum through the software life cycle is given in Appendix F.

3.2 Software quality assurance plan

A quality assurance plan shall exist or be established at an early stage either as a part of the computer system specification (see Clause 4) or as a companion document.

Although the whole of this standard and its appendices can be considered as a quality assurance plan for safety related software, special quality assurance plans may be adopted for individual product phases or particular software components according to national standards or company regulations.

These plans should address all quality assurance procedures required during all phases of the software life cycle.

4. Software requirements

The software requirements (F.M2) shall be derived from requirements of the safety systems and are part of the computer system specification. The computer system specification is a description of the combined hardware/software system and states the objectives and functions assigned to the computer system.

The software requirements describe the product, not the project. They shall describe what has to be done and not how it has to be done. An acceptable approach to the development and content of the software requirements, which is consistent with this clause, is given in Appendix A.

Whereas the main purpose of the software requirements document is to form the basis for software development, the licensing aspects should not be neglected. Therefore, it may

négligés. En conséquence, il peut contenir des éléments moins importants pour la conception du logiciel qui sont cependant des informations de base pour les organismes de sûreté. Ce sont par exemple:

- les considérations liées aux risques;
- des recommandations pour les fonctions des systèmes de sauvegarde et de sûreté;
- d'autres éléments qui fournissent des bases pour les exigences spécifiques.

Les exigences fonctionnelles relatives au logiciel détaillent les fonctions du système qui seront mises en œuvre par le système programmé.

Les exigences en matière de fiabilité du logiciel détaillent, en ce qui concerne le logiciel, les exigences en fiabilité du système programmé. Elles doivent en être issues de la même manière que les exigences fonctionnelles.

- 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).
- 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).
- 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.

Ce dialogue homme-machine doit être spécifié en tenant compte:

- de la recette du système et de sa mise en service;
- de l'exploitation du système;
- de la maintenance du système et des essais après mise en service (A2.3).

- 4.4 L'interface entre le système programmé et tout autre système à l'intérieur ou à l'extérieur de la centrale nucléaire, qu'une connexion directe existe ou soit prévue, doit être identifiée et documentée en indiquant les interfaces spécifiques et les exigences pour le logiciel qui en découlent (Réf. Publication 639 de la CEI) (A2.4).
- 4.5 Les fonctions destinées à prévenir une émission de radioactivité pour les conditions accidentelles (A2.1) et les fonctions associées doivent être décrites (A2.5). Il convient de décrire les fonctions à réaliser plutôt que la manière de les mettre en œuvre.
- 4.6 Les contraintes entre matériel et logiciel doivent être décrites (A2.6). La référence au document de spécifications du matériel doit être indiquée.
- 4.7 Les conditions particulières de fonctionnement, tels le démarrage de la centrale et le rechargement en combustible, doivent être décrites (A2.7).
- 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.
 - 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;
 - 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.
 - 4.8.2 Le logiciel du système de sûreté doit être conçu de manière à ce que les fonctions essentielles de l'ensemble du système de sûreté puissent être testées durant le fonctionnement de la centrale nucléaire.

contain aspects of minor importance to software design which are, however, a background for licensing. Such aspects may be:

- risk considerations;
- recommendations for functions or engineered safety features;
- other items that provide the background for specific requirements.

The software functional requirements represent an expansion of the functions which are assigned to the computer system and which are to be implemented by the computer system.

The software reliability requirements represent an expansion of the reliability requirements of the computer system. They shall be derived in a similar way to the functional 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).
- 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).
- 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 Sub-clauses 7.12 and 7.3.2 are applicable.

This man-machine dialogue shall be specified with regard to:

- commissioning of the system and start up;
- system operation;
- system maintenance, and testing after commissioning (A2.3).

- 4.4 The interface between the computer system and any other system either within or outside the nuclear plant, wherever a direct connection exists or is planned, shall be identified and documented indicating the specific interfaces and the related software requirements (Ref. IEC Publication 639) (A2.4).
- 4.5 Functions dedicated to the prevention of the release of radioactivity under accident conditions (A2.1) and their related system functions shall be described (A2.5). The functions to be performed should be defined rather than the means of implementation.
- 4.6 The constraints between hardware and software shall be described (A2.6). A reference to the hardware requirements document shall be made.
- 4.7 Special operating conditions such as plant commissioning and refuelling shall be described (A2.7).
- 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.
 - 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;
 - b) those parts of the memory that contain code or invariable data shall be monitored to prevent any changes.
 - 4.8.2 The safety system software shall be designed in such a manner that essential functions of the whole safety system be testable during the operation of the nuclear power generating station.

- 4.8.3 Si une panne quelconque du système de sûreté est détectée durant le fonctionnement de la centrale, les actions automatiques appropriées doivent être prises en tenant compte des exigences temporelles. Cela peut impliquer de prendre en considération des dispositions pour éviter des déclenchements intempestifs.
- 4.8.4 Les autocontrôles du système ne doivent pas dégrader les fonctions qu'il doit remplir.
- 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.
 - b) Toute dégradation dans l'exécution des fonctions de sûreté doit être détectée.
 - 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.
- 4.9 Les exigences fonctionnelles pour le logiciel doivent être présentées selon un modèle dont le formalisme ne doit pas, en principe, nuire à la compréhension (A2.9). Les exigences doivent être non équivoques, testables ou vérifiables, et réalisables.
- 4.10 L'utilisation d'un langage formel de spécification peut être une aide pour démontrer la cohérence et l'exhaustivité des exigences fonctionnelles pour le logiciel. Des outils automatiques peuvent être utilisés à cet effet.
- 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;
 - au responsable de l'équipe de conception du logiciel pour accord, en ce qui concerne les aspects faisabilité et lisibilité;
 - aux responsables de la sûreté pour autorisation d'exploitation, en ce qui concerne le respect de l'ensemble des exigences de sûreté du réacteur.

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

Les recommandations ci-après fournissent un guide de bonne pratique pour écrire un logiciel aussi exempt d'erreur que possible dès le début de la conception et pouvant être aisément vérifié.

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.

5.1 Principes du développement (conception et codage)

- 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.
 - Il convient que la structure du programme soit basée sur une décomposition en modules.
 - 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.
 - Il convient que le programme source résultat soit compréhensible dans sa totalité.
 - Une bonne documentation doit être fournie.

- 4.8.3 If any failure is detected during plant operation, appropriate and timely automatic actions shall be taken. This may require giving due consideration to avoiding spurious trips.
- 4.8.4 System self-checking shall not adversely affect the intended system functions.
- 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.
 - b) Any degradation of the execution of safety functions shall be detected.
 - 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 Sub-clause 10.3.

- 4.9 Software functional requirements shall be presented according to a standard whose formality should not preclude readability (A2.9). The requirements shall be unequivocal, testable or verifiable, and realizable.
- 4.10 The use of a formal specification language may be a help to show coherence and completeness of the software functional requirements. Automatic tools may be used for this purpose.
- 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;
 - to the software design team manager for approval and with respect to feasibility and readability;
 - to the licensors with respect to compliance with the overall plant safety requirements for licensing approval.

5. Development (design and coding) of safety system software

The following recommendations provide a guide to good practice for writing software which is as error-free as possible from the very beginning and which can easily be verified.

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

5.1 Principles for development (design and coding)

- 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 Sub-clause 4.8.
 - The program structure should be based on a decomposition into modules.
 - 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 final source program should be readable from start to end.
 - Good documentation shall be provided.

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);
- b) une approche descendante soit préférée à une approche ascendante dans le développement du logiciel (B1);
- c) un modèle conceptuel de la structure du système soit adopté au début de chaque projet logiciel (B2);
- d) le programme soit écrit pour être facilement testable (B4, B5);
- e) dans le cas d'utilisation d'un logiciel commercialisé, la démonstration de son bon fonctionnement soit réalisée (B2.c).

5.2 Langage, traducteur et autres outils

Des directives relatives au choix des langages, traducteurs, etc., sont données dans l'annexe D.

Bien qu'un langage spécifique ne puisse être requis, ce qui suit peut être considéré comme les règles de base communes pour les langages de programmation de système de sûreté.

- 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.
- 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.
- 5.2.3 Les langages orientés «application» sont fortement préférables aux langages orientés «machine».
- 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.
- 5.2.5 Il convient que des outils automatiques de test soient disponibles.
- 5.2.6 Il est recommandé d'utiliser des outils automatiques.

5.3 Recommandations détaillées

5.3.1 Développement

Dans l'annexe B figure une série de recommandations qui spécifient en détail les aspects du paragraphe 5.1.

Les têtes de chapitre des recommandations individuelles de l'annexe B, ainsi que les articles appropriés de la présente norme, peuvent aussi être affectés aux deux parties essentielles du développement d'un programme, la conception et le codage, suivant le tableau ci-après.

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);
- b) a top down approach to software development should be preferred to a bottom up approach (B1);
- c) a conceptual model of system structure should be adopted at the beginning of each software project (B2);
- d) the program should be written to allow easy testing (B4, B5);
- e) where standard software from a manufacturer or supplier is used, it should be shown to have operated satisfactorily (B2.c).

5.2 *Language, translator and other tools*

Guidance for selection of language, translator, etc., is given in Appendix D.

Even though a specific programming language cannot be required, the following may be considered as common basic rules for safety system programming languages.

- 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.
- 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.
- 5.2.3 Problem oriented languages are strongly preferred to machine oriented ones.
- 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.

5.2.5 Automatic testing aids should be available.

5.2.6 The use of automatic tools is recommended.

5.3 *Detailed recommendations*

5.3.1 *Development*

In Appendix B a set of recommendations is given which specifies in detail the aspects identified in Sub-clause 5.1.

The headings of the individual recommendations of Appendix B as well as relevant clauses of this standard can also be attributed to the two major parts of program development, design and coding, according to the following table.

Aspects, procédures et structures de conception et de codage de programmes d'application pour les systèmes de sûreté

	Aspects procédures	Rubrique/ Article	Aspects structures	Rubrique/ Article
Conception	Adaptabilité Approche descendante Vérification intermédiaire Modifications en cours de développement Reconfiguration du système	1.a/4. 1.b 1.c/6. 1.d/9. 1.e/9.	Structure de contrôle et d'accès Modules Systèmes d'exploitation Temps d'exécution Interruptions Expressions arithmétiques Contrôle de vraisemblance Sûreté des sorties Branchements et boucles Sous-programmes et procédures Structures imbriquées Structures des données	2.a 2.b 2.c 2.d/4. 2.e 2.f 3.a/4. 3.b/4. 4.a 4.b 4.c 4.e
Codage	Vérification intermédiaire Modification en cours de développement Tests intermédiaires Règles de codage	1.c/6. 1.d/9. 4.g/6. 5.d	Modules Temps d'exécution Interruptions Expressions arithmétiques Contrôle de vraisemblance Sûreté des sorties Contenu mémoire Contrôle des erreurs Branchements et boucles Sous-programmes et procédures Structures imbriquées Adressage et zones de données Structures des données Modifications dynamiques Organisation du programme Commentaires Assembleur	2.b 2.d/4. 2.e 2.f 3.a/4. 3.b/4. 3.c 3.d 4.a 4.b 4.c 4.d 4.e 4.f 5.a 5.b 5.c

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.

En liaison avec certains problèmes particuliers, il peut être raisonnable de modifier les recommandations contenues dans l'annexe B afin d'atteindre l'objectif de simplicité et de compréhension de l'ensemble. Ces modifications doivent être documentées, en tenant compte de ce qui suit:

- durant la procédure de sélection des recommandations, il convient de considérer les parties du logiciel relevant particulièrement de la sûreté;
- pour les sections du logiciel les plus critiques du point de vue de la sûreté du système, il y a lieu de sélectionner et d'appliquer les recommandations les plus contraignantes;
- il est recommandé que les sections de programme relevant particulièrement de la sécurité soient clairement identifiées dans la structure du système et des données utilisées;

Procedural and structural aspects of design and coding of user program for safety system

	Procedural aspects	Item/ Clause	Structural aspects	Item/ Clause
Design	Changeability	1.a/4.	Control and access structures	2.a
	Top-down approach	1.b	Modules	2.b
	Intermediate verification	1.c/6.	Operating system	2.c
	Changes during development	1.d/9.	Execution time	2.d/4.
	System reconfiguration	1.e/9.	Interrupts	2.e
			Arithmetic expressions	2.f
			Plausibility check	3.a/4.
			Safe output	3.b/4.
			Branches and loops	4.a
			Subroutines and procedures	4.b
			Nested structures	4.c
			Data structures	4.e
Coding	Intermediate verification	1.c/6.	Modules	2.b
	Changes during development	1.d/9.	Execution time	2.d/4.
	Intermediate tests	4.g/6.	Interrupts	2.e
	Coding rules	5.d	Arithmetic expressions	2.f
			Plausibility check	3.a/4.
			Safe output	3.b/4.
			Memory contents	3.c
			Error checking	3.d
			Branches and loops	4.a
			Subroutines and procedures	4.b
			Nested structures	4.c
			Addressing and arrays	4.d
			Data structures	4.e
			Dynamic changes	4.f
			Sequences and arrangements	5.a
			Comments	5.b
			Assembler	5.c

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.

In connection with particular problems it may be reasonable to modify the recommendations selected from Appendix B, in order to meet the goal of overall simplicity and understandability. These modifications shall be documented, taking account of the following:

- during the selection procedure of the recommendations, the safety relevance of particular software parts should be considered;
- the more limiting recommendations should be selected and applied to those program sections which are more critical to the safety of the system;
- program sections of outstanding safety relevance should be clearly identifiable from the system and data structure used;

- d) il y a lieu de tenir compte des facilités de test disponibles et de la stratégie de validation prévue lors de la procédure de sélection. Si des parties importantes sont programmées de manière diversifiée, on peut utiliser pour ces parties une stratégie différente de vérification;
- e) si des difficultés apparaissent durant la phase de vérification, des modifications rétrospectives du style de programmation peuvent être nécessaires.

5.4 *Documentation*

- 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.

- 5.4.2 En plus de l'analyse organique, d'autres documents peuvent être requis. Ils sont utilisés à la fois pour les étapes de vérifications intermédiaires et finales. Certains documents se rapportent à la conception initiale. Ils sont dérivés des spécifications fonctionnelles et constituent la base de l'analyse organique. D'autres sont issus de ce document particulier et servent de référence au codage ultérieur.

Si les recommandations relatives à la procédure de développement donnée dans l'annexe B sont suivies, la documentation appropriée est élaborée automatiquement.

- 5.4.3 Le but est d'obtenir un ensemble intégré de documents. Chaque document doit avoir une relation définie avec les autres documents et traiter d'un sujet bien délimité.

- 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.

6. **Vérification**

6.1 *Processus de vérification*

Après établissement des exigences fonctionnelles du logiciel et avant que ne commence la phase suivante, la vérification consiste à contrôler l'adéquation entre les exigences fonctionnelles pour le logiciel et les exigences du système de sûreté assignées au logiciel par les spécifications de l'équipement programmé.

Après la phase de conception et avant le début de la phase suivante, la vérification consiste à contrôler l'adéquation de la conception du logiciel, incluse dans l'analyse organique, avec les exigences fonctionnelles pour le logiciel (analyse fonctionnelle).

Après la phase de codage et avant le début de la phase suivante, la vérification consiste à contrôler la conformité du code avec l'analyse organique issue de la phase de conception.

Chaque phase doit être complétée par une vérification (A1).

- d) the selection procedure should take into account the available testing facilities and the intended validation strategy. If important parts are programmed diversely, a different verification strategy may be used for those parts;
- e) if difficulties arise during verification, a retrospective change of programming style may be necessary.

5.4 Documentation

- 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 detail 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.

- 5.4.2 In addition to the software performance specification, other documents may be required. They are used for both intermediate and final program verification steps. Some of these documents relate to the first design steps. They are derived from the functional requirements specification and provide the basis for the software performance specification. Others are derived from this particular document and represent the basis for later coding.

If the recommendations for the development procedure according to Appendix B are met, appropriate documentation is provided as a byproduct of program development.

- 5.4.3 The aim is an integrated set of documents. Each document shall have a defined relationship to the other document and contain a well bounded subject-matter.

- 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.

6. Verification

6.1 Software verification process

After the software functional requirements have been established and before the next phase begins, verification addresses the adequacy of the software functional requirements in fulfilling the safety system requirements assigned to the software by the computer system specification.

After the design phase and before the next phase begins, verification addresses the adequacy of computer system software design as documented in the software performance specification to the software functional requirements.

After the coding phase and before the next phase begins, verification addresses the compliance of the coded computer system software to the software performance specification as derived by the design phase.

Each phase shall be completed by verification (A1).

6.2 Activités de vérification du logiciel

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.

Le plan de vérification doit être préparé par une équipe de vérification prenant en charge:

- a) la sélection des stratégies de vérification (systématique et/ou aléatoire avec choix de jeux d'essais selon les fonctions requises et/ou les caractéristiques de la structure du programme, voir annexe E);
- b) la sélection et l'utilisation d'un outil de test du logiciel;
- c) l'exécution de la vérification;
- d) la documentation des activités de vérification;
- 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.

L'encadrement de l'équipe de vérification doit être séparé et indépendant de celui de l'équipement de développement.

6.2.2 Vérification de la conception, revues critiques

La vérification de la conception consiste à examiner:

- a) l'adéquation de l'analyse organique avec les spécifications fonctionnelles du logiciel (analyse fonctionnelle) en ce qui concerne leur cohérence et leur exhaustivité, et cela jusqu'au niveau des modules;
- b) la décomposition de l'analyse organique en modules fonctionnels et la manière selon laquelle ils sont spécifiés, pour les aspects:
 - faisabilité des traitements requis;
 - testabilité pour la vérification ultérieure;
 - lisibilité par les équipes de développement et de vérification;
 - aptitude à la maintenance pour permettre des évolutions ultérieures;
- 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.

6.2.3 Phase de vérification du codage

La vérification du codage commence avec le test des modules et prend en charge le logiciel par une approche ascendante. Au niveau module, le logiciel n'est pas encore intégré dans le système, il peut donc être testé en grande partie. Le but du test des modules est de montrer que chaque module réalise exclusivement la fonction attendue. Un test d'intégration des modules

6.2 Software verification activities

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.

The verification plan shall be prepared by a verification team addressing:

- a) selection of verification strategies, either systematic, random or both, with test case selection according to either required functions, special features of program structure, or both, see Appendix E;
- b) selection and utilization of the software test equipment;
- c) execution of verification;
- d) documentation of verification activities;
- e) evaluation of verification results gained from verification equipment directly and from tests, evaluation of whether the reliability requirements are met.

The management of the verification team shall be separate and independent from the management of the development team.

6.2.2 Design verification, critical reviews

The design verification addresses:

- a) the adequacy of the software performance specification for the software functional requirements with respect to consistency and completeness down to and including the modular level;
- b) the decomposition of the design into functional modules and the manner of specification with reference to:
 - feasibility of the performance required;
 - testability for further verification;
 - readability by the development and verification team;
 - maintainability to permit further evolution;
- 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.

6.2.3 Coding phase verification

The code verification activities begin with module testing and go up through the software by a bottom-up strategy. At the module level, the software is not yet integrated into the system, therefore it can be extensively tested. The purpose of module testing is to show that each module performs its intended function and does not perform unintended functions. A

doit alors être effectué pour montrer au plus tôt l'interaction correcte de tous les modules pour une fonction donnée.

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

6.2.3.1 Spécification de test du logiciel

La spécification de test du logiciel est un des principaux documents du plan de vérification pour la phase de codage. Ce document doit être fondé sur un examen détaillé des exigences fonctionnelles pour le logiciel et doit donner des informations détaillées sur les tests à réaliser pour chaque élément du logiciel (les modules et leurs constituants).

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.

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.

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) configuration du matériel utilisé pour le test;
- b) moyen de stockage et conditions d'accès au code final à tester;
- c) liste des éléments en entrée du test;
- d) liste des éléments en sortie du test;
- e) données supplémentaires en ce qui concerne la chronologie, la séquence d'événements, etc.;
- f) conformité avec les critères d'acceptation donnés dans la spécification du test;
- g) relevé des erreurs décrivant les caractéristiques de chaque erreur et les remèdes mis en œuvre par l'équipe de développement.

7. Intégration matériel/logiciel

Le processus d'intégration du matériel et du logiciel est l'association du matériel contrôlé et du logiciel vérifié dans un système capable de réaliser les fonctions spécifiées. Ce processus comprend quatre parties:

- a) assemblage des sous-ensembles du matériel par câblage selon les plans du système;
- b) assemblage des modules logiciels par un éditeur de liens;

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

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

6.2.3.1 *Software test specification*

The software test specification is one of the principal documents of the verification plan for the coding phase. This document shall be based upon a detailed examination of the software functional requirements and shall give detailed information on the tests to be performed addressing each of the components of the software (modules and their constituents).

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

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.

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:

- a) hardware configuration used for the test;
- b) storage medium used and access requirements of the final code tested;
- c) input test listing;
- d) output test listing;
- e) additional data regarding timing, sequence of events, etc.;
- f) conformance with acceptance criteria given in the test specification;
- g) error incident log which describes the character of the error and the remedies taken by the design team.

7. **Hardware/software integration**

The process of hardware/software integration is the combining of verified hardware and software modules into a system that is capable of performing specified functions. This process consists of four parts:

- a) assembling hardware modules by interconnecting wiring according to the system design drawings;
- b) assembling software modules by a linkage processor;

- c) chargement du logiciel dans le matériel;
- d) vérification par le test que les spécifications d'interface matériel/logiciel sont satisfaites et que le logiciel est capable de fonctionner dans cet environnement matériel.

7.1 *Plan d'intégration du système*

Ce plan doit être préparé et explicité dans les spécifications d'intégration (A1), et vérifié par rapport aux spécifications du système de sûreté. Le plan d'intégration du système (F.M2) décrit les aspects relatifs à l'organisation et aux procédures pour l'intégration matériel/logiciel.

Ce plan doit spécifier les normes et les procédures à suivre lors de l'intégration matériel/logiciel et doit comprendre les dispositions du plan général d'assurance qualité applicables.

Le plan d'intégration du système doit traiter de toutes les contraintes liées à une conception spécifique de matériel et de logiciel. Il doit inclure les exigences pour les procédures et les méthodes de contrôle comprenant:

- la gestion de configuration du système;
- l'intégration du système;
- le test du système intégré;
- le traitement des erreurs.

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.

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.

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.

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;
- la procédure doit s'assurer que l'intégration du système est réalisée en utilisant les bonnes versions des modules individuels;
- la procédure doit fournir l'indice d'utilisation de chaque module du système pour permettre l'évaluation de l'impact des révisions futures;
- la procédure doit être conduite de façon que le plan général d'assurance qualité soit respecté.

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,

- c) loading the software into the hardware;
- d) verifying by testing that the hardware/software interface requirements have been satisfied and that the software is capable of operating in that particular hardware environment.

7.1 *System integration plan*

The system integration plan shall be prepared and documented in the integration requirements (A1) and verified against the safety system requirements. The system integration plan (F.M2) describes the organizational and procedural aspects of the hardware/software integration.

This plan shall specify the standards and procedures to be followed in the hardware/software integration and shall document those provisions of the overall quality assurance plan that are applicable to the system integration.

The system integration plan shall discuss any constraints made by the specific design of the hardware and the software. The plan shall include the requirements for procedures and control methods covering:

- system configuration control;
- system integration;
- integrated system verification testing;
- error resolution.

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.

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.

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.

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;
- the procedure shall ensure that the system integration is performed using the proper revision levels of the individual 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;
- the procedure shall be conducted in such a way that the overall quality assurance plan is met.

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 pro-

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.

Le test réalisé au cours de cette phase d'intégration matériel/logiciel est le test fonctionnel du concepteur du système. C'est l'équivalent au niveau système de la mise au point faite par un programmeur avant vérification. Toutes les erreurs détectées durant ce test, qui sont strictement des erreurs liées à l'intégration elle-même et qui n'affectent aucun document du projet, peuvent être corrigées sans rapport formel de l'erreur. Cependant, si la résolution d'une erreur exige une modification du logiciel vérifié ou de tout autre document de conception, cette erreur doit donner lieu à un rapport selon la procédure du paragraphe 7.6

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.

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.

La vérification du système intégré doit être réalisée selon un plan de test formalisé. Ce plan doit identifier les tests pour chaque exigence relative à l'interface du système programmé. Le test du système intégré doit être réalisé et les résultats évalués par un personnel connaissant les spécifications du système mais n'ayant pas participé à son développement.

Le niveau d'indépendance existant entre les concepteurs et les vérificateurs du système doit être suffisant pour amener toute communication entre les deux parties — soit pour demande de clarification, soit pour constat d'erreur — à se faire uniquement de manière formelle par écrit, afin de pouvoir être vérifié par des personnes qui ne sont pas engagées dans le développement du système.

Les matériels utilisés pour la vérification du système doivent être étalonnés. Des mesures d'assurance qualité doivent être établies pour les outils logiciels utilisés dans la vérification, en rapport avec l'importance de ces outils pour cette action.

7.6 *Procédures de résolution des erreurs*

Des procédures spécifiques pour rendre compte des erreurs et de leur correction doivent être établies en tant qu'éléments du plan d'intégration du système.

Ces procédures doivent être appliquées à toutes les erreurs trouvées au cours de la vérification du système ainsi qu'à celles trouvées pendant les tests fonctionnels ayant nécessité des modifications du logiciel vérifié ou des documents de conception du système. La procédure

cedures 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 Sub-clause 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.

The testing performed in this phase of the hardware/software integration is the designer's functional test of the system. It is the system analogy to the debugging done by a programmer before he releases his software to be verified. Any errors detected during the test that are strictly mistakes in the integration itself, and do not affect any project document, may be corrected without formal error report. However, if the resolution of any error requires a change to verified software or any design document, that error shall be reported according to the procedures established by Sub-clause 7.6.

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.

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.

Integrated system verification shall be conducted in accordance with a formal test plan. The plan shall identify the tests for each computer system interface requirement. The integrated system test shall be developed and the test results evaluated by individuals who are familiar with the system specification and who did not participate in the development of the system.

The level of independence provided between the designers and the system verifiers shall be sufficient to ensure that all communication between the two parties, whether for clarification or error reporting, is only conducted formally in writing at a level of detail which may be audited by persons not involved in the development of the system.

Equipment used for system verification shall be calibrated. Quality assurance measures shall be established for software tools used for verification, commensurate with the importance of those tools for verification.

7.6 Error resolution procedures

Specific procedures for the reporting and resolution of errors shall be prepared as a part of the system integration plan.

These procedures shall apply to all errors found during the system verification as well as those found during the integration functional test that require changes to verified software or system design documents. The procedure shall ensure that any required reverification of

doit donner l'assurance qu'une révérification des modules matériels et/ou logiciels est réalisée et que la nouvelle version des modules a été prise en compte dans la base de données selon les procédures du paragraphe 7.3.

Une évaluation de chaque erreur relevée doit être faite afin de déterminer si elle aurait pu être détectée dans une phase précédente de vérification (par exemple lors du test d'un module). Si cela est le cas, une investigation de cette phase précédente doit être conduite pour déterminer s'il existe une défaillance systématique dans la vérification.

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 résultats incorrects rencontrés en cours de vérification avec les actions correctives entreprises conformément au paragraphe 7.6. Les résultats d'essais 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.

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.

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.

Il est recommandé que les essais:

- couvrent la plage de variation des signaux et les gammes des variables élaborées ou calculées d'une manière pleinement représentative;
- 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.

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 œuvre 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.

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.

hardware or software modules is performed and that the revision to the modules is carried out through the library procedures of Sub-clause 7.3.

An evaluation of each error reported shall be made to determine whether the error was of such a nature that it should have been detected at an earlier phase (such as module testing) of the verification. If this is found to be the case, then an investigation of that earlier stage of the verification shall be conducted to determine whether any systematic deficiency of the verification exists.

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 used and its calibration, the simulation of system or interface components, and any test results discrepancy found along with the corrective actions taken according to Sub-clause 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.

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.

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.

It is recommended that the tests:

- cover all signal ranges, and the ranges of computed or calculated parameters in a fully representative manner;
- cover the voting and other logic and 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.

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.

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.

8.1 *Compte rendu de validation du système programmé*

Les résultats de la validation du système programmé doivent être inclus dans un compte rendu (F.M6). Ce compte rendu doit identifier le matériel et le logiciel utilisés, les outils employés et leur étalonnage, les modèles de simulation utilisés, les anomalies et les actions correctives entreprises selon la procédure de modification de l'article 9.

Le compte rendu doit résumer les résultats de la validation du système programmé et doit montrer comment le système correspond aux spécifications.

Les résultats doivent être présentés de sorte qu'ils puissent être vérifiés par les personnes qui n'ont pas été directement impliquées dans la validation. Il est recommandé de citer dans le compte rendu les outils logiciels utilisés dans le processus de validation. Toute simulation de la centrale et de ses systèmes, utilisée pour la validation, doit être mentionnée.

9. **Maintenance et modification**

Une procédure formelle de gestion des modifications doit être établie, incluant la vérification et la validation.

9.1 *Procédure et demande de modification*

Pour qu'une modification puisse être considérée, la procédure suivante doit être respectée.

9.1.1 Toute demande de modification doit être établie en indiquant:

- sa raison d'être;
- son but;
- son domaine fonctionnel;
- l'identité du demandeur.

La modification peut être demandée pendant la phase de développement à la suite d'un changement dans les spécifications fonctionnelles causé par:

- une modification fonctionnelle;
- une évolution technologique;
- un changement dans les conditions d'exploitation.

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.

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 œuvre 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.

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

- la faisabilité;
- 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;

8.1 *Computer system validation report*

The results of the computer system validation shall be documented in a report (F.M6). This report shall identify the hardware and software used, the equipment used and its calibration, the simulation models used, and any discrepancies and corrective actions according to the modification procedure of Clause 9.

The report shall summarize the results of the computer system validation and shall assess the system compliance with all requirements.

The results shall be retained in a form that is auditable by persons not directly engaged in the validation. Software tools used in the validation process should be identified as an item in the validation report. Simulations of the plant and its systems used for the validation shall be documented.

9. **Maintenance and modification**

A formal modification control procedure shall be established including verification and validation.

9.1 *Modification request procedure*

For a modification to be considered, the following procedure shall be followed.

9.1.1 A software modification request shall be generated, and uniquely identified stating:

- reason for request;
- aim;
- functional scope;
- originator.

The modification may be requested during the development phase because of a change of functional requirements caused by:

- functional modification;
- technological evolution;
- changes of operating conditions.

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.

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.

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

- technical feasibility;
- 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;

- les conséquences sur les performances (temps de calcul, précision, etc.);
 - l'effort nécessaire de vérification et de validation; l'analyse de la vérification ultérieure nécessaire du logiciel doit être explicitée de manière à permettre un audit;
 - la liste des documents à revoir.
- 9.1.4 La demande d'une modification de logiciel reste en suspens jusqu'à ce qu'une décision soit prise pour:
- l'accepter (immédiatement ou après l'examen du compte rendu d'analyse des incidences) et la mettre en œuvre, ou
 - la rejeter en le justifiant.

9.2 *Procédure d'exécution d'une modification*

- 9.2.1 Pour les modifications sur le site, il convient que le fournisseur du logiciel ait accès à une configuration d'essai identique au système réel dans tous ses aspects significatifs (matériel, compilateur, outils de tests, simulateur de la centrale, etc.) pour assurer la validité des modifications.
- 9.2.2 La procédure de modification dépend du niveau où elle est introduite:
- un changement de spécification fonctionnelle exige un nouvel examen de toute la procédure de développement du logiciel, en ce qui concerne la partie du système touchée par la modification;
 - un changement pendant la phase de développement doit être contrôlé en ce qui concerne les incidences sur les niveaux les plus bas correspondants;
 - les modifications doivent être traitées selon les règles figurant dans l'article 5.
- 9.2.3 Après réalisation d'une modification, le processus complet ou partiel de vérification et de validation décrit au paragraphe 6.1 et dans l'article 8 doit être effectué selon l'analyse des incidences de la modification (voir paragraphe 9.1.3).
- 9.2.4 Tous les documents concernés par la modification doivent être corrigés et faire référence à l'identification de la demande de modification logicielle. Un compte rendu de modification logicielle doit résumer toutes les actions faites à son sujet.
- Tous ces documents doivent être datés, numérotés et classés dans l'historique de modification du logiciel du projet.

9.3 *Maintenance*

Les raisons d'une maintenance peuvent être:

- un compte rendu d'anomalie;
 - une modification des spécifications fonctionnelles après livraison;
 - une évolution technologique;
 - un changement dans les conditions d'exploitation.
- 9.3.1 En cas d'anomalie, un compte rendu d'anomalie doit être établi donnant les symptômes, l'environnement et l'état du système lors de la découverte de l'incident, ainsi que les causes suspectées. La correction de l'anomalie requiert l'élaboration d'une demande de modification qui doit suivre la procédure décrite au paragraphe 9.1.
- 9.3.2 Dans le cas de modification des spécifications fonctionnelles ou des conditions d'exploitation, tout le processus de développement du logiciel doit être repris, en ce qui concerne la partie du système touchée par la modification.
- 9.3.3 De nouvelles exigences et possibilités du matériel doivent être évaluées en ce qui concerne leur impact potentiel sur le logiciel. Pour cette évaluation, il convient de prendre en considération les caractéristiques du matériel examinées lors de la conception initiale du logiciel. S'il

- impact upon performance (including speed, accuracy, etc.);
- necessary effort for verification and validation; the analysis of the software reversion needed shall be documented in an auditable form;
- the set of documents to be reviewed.

9.1.4 The software modification request is pending until the decision is made:

- to accept it (immediately, or after examination of the software modification analysis report) and to execute it, or
- to reject it and justify the rejection.

9.2 *Procedure for executing a modification*

9.2.1 For modifications on site, the software supplier should have access to a test configuration which is identical to the real system in all relevant aspects (including installed machine, translator, testing tools, plant simulator, etc.) to ensure the validity of the modifications.

9.2.2 A modification procedure depends upon the level at which it is introduced:

- for a change of the software functional specification, the whole software development process for any part of the system impacted by the change shall be re-examined;
- a change during the development phase shall be reviewed in terms of its potential impact upon corresponding lower levels;
- the modification shall be carried out according to the rules given in Clause 5.

9.2.3 After implementation of the modification, the whole or part of the verification and validation process described in Sub-clause 6.1 and Clause 8, shall be performed again according to the software modification analysis (see Sub-clause 9.1.3).

9.2.4 All the documents affected by the modification shall be corrected and refer to the identification of the software modification request. A software modification report shall sum up all the actions made for modification purposes.

All these documents shall be dated, numbered and filed in the software modification control history for the project.

9.3 *Maintenance*

The reasons for maintenance include:

- an anomaly report;
- a functional requirements change after delivery;
- technological evolution;
- a change in operating conditions.

9.3.1 In case of an anomaly, an anomaly report shall be written giving the symptoms, the system environment and system status at the time at which the anomaly was discovered and the suspected causes. Anomaly correction requires the generation of a software modification request, the execution of which shall follow the procedure described in Sub-clause 9.1.

9.3.2 In case of a change in software functional requirements or in operating conditions, the whole software development process shall be re-examined for that part of the system impacted by the change.

9.3.3 Any new hardware requirements and capabilities shall be examined with respect to their potential impact on the software systems. This evaluation should include all hardware considerations reviewed in the original software design. If it can be shown that the new system

est possible de montrer que la nouvelle configuration matériel du système n'a pas d'incidence sur les spécifications du logiciel, une procédure simplifiée peut être suivie pour mettre en œuvre la modification soit au niveau de la conception, soit au niveau du codage.

- 9.3.4 Dans tous les cas, après implantation de la modification sur l'équipement du site, on doit établir un «document site» donnant des informations sur la date de l'implantation et les résultats des observations effectuées. Ce document doit être classé dans l'historique de modification du logiciel relatif au projet.

10. Exploitation

Le terme exploitation se rapporte aux activités-concernant la mise à disposition et le fonctionnement du système. Il recouvre le système programmé et le personnel d'exploitation.

10.1 *Système programmé*

10.1.1 *Essais de mise à disposition*

Un programme d'essai doit être fourni pour vérifier l'intégrité du système de sûreté programmé, en ce qui concerne les temps de réponse, les réglages, les opérations fonctionnelles et les interactions avec les autres systèmes.

Un programme d'essais de mise à disposition comprenant les critères d'acceptation, les jeux d'essais et l'environnement des essais doit être établi (F.M6).

Un compte rendu de mise à disposition présentant les résultats d'essai et attestant le respect des critères d'essai doit être établi (F.M7).

10.1.2 *Interaction homme-machine*

Pour permettre l'interaction homme-machine, des fonctions de commande opérateur du système programmé peuvent être requises. Les périphériques utilisés à cet effet ne doivent pas fournir la possibilité d'altérer le programme logique mémorisé du système programmé.

Une procédure appropriée et/ou un verrouillage doivent être établis pour prévenir le changement par inadvertance par l'opérateur de paramètres qui peuvent affecter les points de consigne du système de sécurité.

Toutes les actions réalisées par l'opérateur à travers les commandes opérateur doivent être consignées par des moyens appropriés.

10.2 *Formation des opérateurs*

10.2.1 *Programme de formation*

Afin d'obtenir une exploitation sûre de la centrale, le comportement de l'exploitant est aussi important que la fiabilité de l'équipement. Un programme de formation des opérateurs doit donc être fourni à la fois aux exploitants de la centrale et aux spécialistes de l'instrumentation et de la commande, formation en rapport avec la complexité des systèmes de protection mis en œuvre. Le programme de formation doit traiter des conditions de fonctionnement aussi bien normales qu'anormales. Toutes les interfaces du système de protection programmé avec les exploitants doivent être incluses dans la formation. Il est recommandé qu'une formation spécifique à la reconnaissance des anomalies matériel et logiciel soit aussi incluse dans le programme.

10.2.2 *Plan*

Un plan de formation homogène avec les principes du programme de formation doit être établi (F.M6).

Un manuel utilisateur doit être disponible pour l'exploitant du réacteur. Il convient que ce manuel définisse chaque élément d'interface opérateur. Chaque fonction de chaque dispositif doit être expliquée et illustrée selon sa complexité.

does not impact the software requirements, a simplified procedure may be used to implement the modification either at the design or coding phase.

- 9.3.4 In all cases, after implementation of the modification upon the in-the-field equipment, a field document shall be issued which gives the date of the implementation and the result of the specified observations. This document shall be filed in the software modification control history for the project.

10. Operation

The term 'operation' is used to describe all activities regarding the commissioning and the operation of the system. It concerns the computer system and the operating personnel.

10.1 Computer system

10.1.1 Commissioning tests

A test program shall be provided to verify the integrity of the installed computer-based safety system with respect to response, calibration, functional operation and interaction with other systems.

A commissioning test plan, consisting of acceptance criteria, test cases and test environment shall be established (F.M6).

A commissioning test report presenting test results and conformity to acceptance criteria shall be established (F.M7).

10.1.2 Man-machine interaction

To allow man-machine interaction, operator control of computer functions may be required. These devices shall not provide the capability to alter the stored computer program logic.

An appropriate procedure and/or locking device shall be established to prevent the operator inadvertently changing parameters that can affect the set points of the safety system.

All the actions performed by the operator through the operator control shall be documented by suitable means.

10.2 Operator training

10.2.1 Training program

In order to achieve safe plant operation, operator behaviour is as important as equipment reliability. Therefore an operator training program shall be provided both for plant operators and instrumentation and control specialists consistent with the complexity of the protective functions implemented. The training program shall provide for normal and abnormal reactor operation. All operator interfaces of the computer system shall be included in the training program. Specific training in the recognition of hardware and software abnormalities should also be included in the program.

10.2.2 Training plan

A training plan consistent with the principles of the training program shall be established (F.M6).

A user manual shall be provided for the reactor operator (F.M6). The user manual should define each operator interface device. Each function of each device shall be explained and illustrated in accordance with its complexity.

10.2.3 *Système de formation*

La formation des exploitants doit être conduite sur un système de formation qui est équivalent au véritable système matériel/logiciel. La simulation du réacteur pour ce système doit être réalisée par un système de test capable de simuler les conditions normales et anormales du réacteur.

Il convient de plus que ce système permette la formation à chaque interface opérateur. Un simulateur programmé peut être utilisé à cet effet.

10.3 *Spécifications du logiciel relatives aux essais périodiques*

Les principes généraux de la Publication 671 de la CEI: Essais périodiques et surveillance du système de protection des réacteurs nucléaires, sont applicables pour les tests périodiques.

De plus, les exigences suivantes non couvertes par le paragraphe 4.8 doivent être respectées.

10.3.1 *Programme d'essais*

Un programme pour les essais périodiques des systèmes de sûreté doit être défini et inclure les essais fonctionnels applicables, les appareils, les tests, la vérification des réglages et des temps de réponse.

10.3.2 *Spécification des essais*

Les essais doivent vérifier les aptitudes fonctionnelles de base du logiciel, comprenant:

- toutes les fonctions de sûreté de base devant être périodiquement testées;
- des essais spéciaux prévus pour détecter les défauts qui ne peuvent être décelés par les dispositifs d'auto-test incorporés dans le système de sûreté ou par les indicateurs d'anomalies ou d'alarmes;
- les fonctions importantes ne se rapportant pas à la sûreté et qu'il convient de tester périodiquement. Il convient d'effectuer un essai capable de détecter une dégradation de ces fonctions.

10.3.3 *Exigences relatives aux systèmes auxiliaires*

Les logiciels des systèmes auxiliaires utilisés pour les essais périodiques des systèmes de sûreté ne sont pas tenus de satisfaire à toutes les exigences du logiciel de sûreté. Il est recommandé que leur qualité corresponde à celle des outils de vérification mentionnée dans l'article 8. Il convient que les données des tests soient choisies selon le tableau E4.2 de l'annexe E.