COLLECTION NORMES DE SÛRETÉ DE L'AIEA

Logiciels destinés aux systèmes programmés importants pour la sûreté des centrales nucléaires

GUIDE DE SÛRETÉ

Nº NS-G-1.1

IAEA
Agence internationale de l'énergie atomique

PUBLICATIONS DE L'AIEA RELATIVES À LA SÛRETÉ

NORMES DE SÛRETÉ

En vertu de l'article III de son Statut, l'AIEA a pour attributions d'établir des normes de sûreté pour la protection contre les rayonnements ionisants et de prendre des dispositions pour l'application de ces normes aux activités nucléaires pacifiques.

Les publications concernant la réglementation par lesquelles l'AIEA établit des normes et des mesures de sûreté paraissent dans la **collection Normes de sûreté de l'AIEA**. Cette collection couvre la sûreté nucléaire, la sûreté radiologique, la sûreté du transport et la sûreté des déchets, ainsi que la sûreté générale (c'est-à-dire intéressant plusieurs de ces quatre domaines), et comporte les catégories suivantes: **fondements de sûreté, prescriptions de sûreté** et **guides de sûreté**.

Les **fondements de sûreté** (lettrage bleu) présentent les objectifs, les notions et les principes fondamentaux de sûreté et de protection pour le développement et l'application de l'énergie nucléaire à des fins pacifiques.

Les **prescriptions de sûreté** (lettrage rouge) établissent les prescriptions qui doivent être respectées pour assurer la sûreté. Ces prescriptions, énoncées au présent de l'indicatif, sont régies par les objectifs et les principes présentés dans les fondements de sûreté.

Les **guides de sûreté** (lettrage vert) recommandent les mesures, conditions ou procédures permettant de respecter les prescriptions de sûreté. Les recommandations qu'ils contiennent sont énoncées au conditionnel pour indiquer qu'il est nécessaire de prendre les mesures recommandées ou des mesures équivalentes pour respecter les prescriptions.

Les normes de sûreté de l'AIEA n'ont pas force obligatoire pour les États Membres, mais ceux-ci peuvent, à leur discrétion, les adopter pour application, dans le cadre de leur réglementation nationale, à leurs propres activités. L'AIEA est tenue d'appliquer les normes à ses propres opérations et aux opérations pour lesquelles elle fournit une assistance.

Pour obtenir des renseignements sur le programme de normes de sûreté de l'AIEA (y compris sur les éditions dans d'autres langues que l'anglais), il convient de consulter le site Internet de l'AIEA à l'adresse suivante :

www.iaea.org/ns/coordinet

ou de s'adresser à la Section de la coordination en matière de sûreté, AIEA, B.P. 100, A-1400 Vienne (Autriche).

AUTRES PUBLICATIONS CONCERNANT LA SÛRETÉ

En vertu de l'article III et du paragraphe C de l'article VIII de son Statut, l'AIEA favorise l'échange d'informations sur les activités nucléaires pacifiques et sert d'intermédiaire entre ses États Membres à cette fin.

Les rapports sur la sûreté et la protection dans le cadre des activités nucléaires sont publiés dans d'autres collections, en particulier la **collection Rapports de sûreté de l'AIEA**, à des fins d'information. Ces rapports peuvent décrire les bonnes pratiques, donner des exemples concrets et proposer des méthodes détaillées pour respecter les prescriptions de sûreté. Ils n'établissent pas de prescriptions et ne contiennent pas de recommandations.

Les autres collections de l'AIEA dans lesquelles sont publiés des documents destinés à la vente concernant la sûreté sont les suivantes : collection Rapports techniques, **collection Rapports d'évaluation radiologique** et **collection INSAG**. L'AIEA édite aussi des rapports sur les accidents radiologiques et d'autres publications spéciales destinées à la vente. Les publications gratuites concernant la sûreté paraissent dans les collections **Documents techniques (TECDOC)** et **Cours de formation**, et en anglais uniquement dans les collections **Provisional Safety Standards Series, IAEA Services Series, Computer Manual Series** et **Practical Radiation Safety and Protection Manuals**.

LOGICIELS DESTINÉS AUX SYSTÈMES PROGRAMMÉS IMPORTANTS POUR LA SÛRETÉ DES CENTRALES NUCLÉAIRES

Les États ci-après sont Membres de l'Agence internationale de l'énergie atomique:

AFGHANISTAN
AFRIQUE DU SUD
ALBANIE
ALGÉRIE
ALLEMAGNE
ANGOLA
ARABIE SAOUDITE
ARGENTINE
ARMÉNIE
AUSTRALIE
AUTRICHE
AZERBAÏDJAN
BANGLADESH
BÉLARUS
BELGIQUE
BÉNIN
BOLIVIE
BOSNIE-HERZÉGOVINE
BOTSWANA
BRÉSIL
BULGARIE
BURKINA FASO
CAMEROUN
CANADA
CHILI
CHINE
CHYPRE
COLOMBIE
CORÉE, RÉPUBLIQUE DE
COSTA RICA
CÔTE D'IVOIRE
CROATIE
CUBA
DANEMARK
ÉGYPTE
EL SALVADOR
ÉMIRATS ARABES UNIS
ÉQUATEUR
ÉRYTHRÉE
ESPAGNE
ESTONIE
ÉTATS-UNIS D'AMÉRIQUE
ÉTHIOPIE
FÉDÉRATION DE RUSSIE
FINLANDE
FRANCE
GABON
GÉORGIE
GHANA
GRÈCE
GUATEMALA
HAÏTI
HONDURAS
HONGRIE
ILES MARSHALL
INDE
INDONÉSIE
IRAN, RÉP. ISLAMIQUE D'
IRAQ
IRLANDE
ISLANDE
ISRAËL
ITALIE
JAMAHIRIYA ARABE LIBYENNE
JAMAÏQUE
JAPON
JORDANIE
KAZAKHSTAN
KENYA
KIRGHIZISTAN
KOWEÏT
LETTONIE
L'EX-RÉPUBLIQUE YOUGOSLAVE DE MACÉDOINE
LIBAN
LIBÉRIA
LIECHTENSTEIN
LITUANIE
LUXEMBOURG
MADAGASCAR
MALAISIE
MALI
MALTE
MAROC
MAURICE
MEXIQUE
MONACO
MONGOLIE
MYANMAR
NAMIBIE
NICARAGUA
NIGER
NIGERIA
NORVÈGE
NOUVELLE-ZÉLANDE
OUGANDA
OUZBÉKISTAN
PAKISTAN
PANAMA
PARAGUAY
PAYS-BAS
PÉROU
PHILIPPINES
POLOGNE
PORTUGAL
QATAR
RÉPUBLIQUE ARABE SYRIENNE
RÉPUBLIQUE CENTRAFRICAINE
RÉPUBLIQUE DÉMOCRATIQUE DU CONGO
RÉPUBLIQUE DE MOLDOVA
RÉPUBLIQUE DOMINICAINE
RÉPUBLIQUE TCHÈQUE
RÉPUBLIQUE-UNIE DE TANZANIE
ROUMANIE
ROYAUME-UNI DE GRANDE-BRETAGNE ET D'IRLANDE DU NORD
SAINT-SIÈGE
SÉNÉGAL
SERBIE ET MONTÉNÉGRO
SEYCHELLES
SIERRA LEONE
SINGAPOUR
SLOVAQUIE
SLOVÉNIE
SOUDAN
SRI LANKA
SUÈDE
SUISSE
TADJIKISTAN
THAÏLANDE
TUNISIE
TURQUIE
UKRAINE
URUGUAY
VENEZUELA
VIETNAM
YÉMEN
ZAMBIE
ZIMBABWE

Le Statut de l'Agence a été approuvé le 23 octobre 1956 par la Conférence sur le Statut de l'AIEA, tenue au Siège de l'Organisation des Nations Unies, à New York; il est entré en vigueur le 29 juillet 1957. L'Agence a son Siège à Vienne. Son principal objectif est «de hâter et d'accroître la contribution de l'énergie atomique à la paix, la santé et la prospérité dans le monde entier».

Imprimé par l'AIEA en Autriche
Février 2004
STI/PUB/1095

COLLECTION DES NORMES DE SÛRETÉ N° NS-G-1.1

LOGICIELS DESTINÉS AUX SYSTÈMES PROGRAMMÉS IMPORTANTS POUR LA SÛRETÉ DES CENTRALES NUCLÉAIRES

GUIDE DE SÛRETÉ

AGENCE INTERNATIONALE DE L'ÉNERGIE ATOMIQUE
VIENNE, 2004

CE VOLUME DE LA COLLECTION SÉCURITÉ EST PUBLIÉ ÉGALEMENT
EN ANGLAIS, EN CHINOIS, EN ESPAGNOL ET EN RUSSE.

LOGICIELS DESTINÉS
AUX SYSTÈMES PROGRAMMÉS
IMPORTANTS POUR LA SÛRETÉ
DES CENTRALES NUCLÉAIRES

AIEA, VIENNE, 2004
STI/PUB/1095
ISBN 92–0–202004–3
ISSN 1020–5829

AVANT-PROPOS

par Mohamed ElBaradei
Directeur général

Une des fonctions statutaires de l'AIEA est d'établir ou d'adopter des normes de sûreté destinées à protéger la santé, les personnes et les biens dans le cadre du développement et de l'utilisation de l'énergie nucléaire à des fins pacifiques et de prendre des dispositions pour appliquer ces normes à ses propres opérations, ainsi qu'à celles pour lesquelles elle fournit une assistance et, à la demande des parties, aux opérations effectuées en vertu d'un accord bilatéral ou multilatéral ou, à la demande d'un État, à telle ou telle des activités de cet État dans le domaine de l'énergie nucléaire.

Les organes consultatifs ci-après supervisent l'élaboration des normes de sûreté : Commission consultative pour les normes de sûreté (ACSS), Comité consultatif pour les normes de sûreté nucléaire (NUSSAC), Comité consultatif pour les normes de sûreté radiologique (RASSAC), Comité consultatif pour les normes de sûreté relatives au transport (TRANSSAC) et Comité consultatif pour les normes de sûreté relatives aux déchets (WASSAC). Les États Membres sont largement représentés au sein de ces comités.

Afin que les normes de sûreté puissent faire l'objet du consensus le plus large possible, elles sont aussi soumises à tous les États Membres pour observation avant d'être approuvées par le Conseil des gouverneurs de l'AIEA (fondements de sûreté et prescriptions de sûreté) ou par le Comité des publications au nom du Directeur général (guides de sûreté).

Les normes de sûreté de l'AIEA n'ont pas force obligatoire pour les États Membres, mais ceux-ci peuvent, à leur discrétion, les adopter pour application, dans le cadre de leur réglementation nationale, à leurs propres activités. L'AIEA est tenue de les appliquer à ses propres opérations et à celles pour lesquelles elle fournit une assistance. Tout État souhaitant conclure un accord avec l'AIEA en vue d'obtenir son assistance pour le choix du site, la conception, la construction, les essais de mise en service, l'exploitation ou le déclassement d'une installation nucléaire ou toute autre activité est tenu de se conformer aux parties des normes qui se rapportent aux activités couvertes par l'accord. Quoi qu'il en soit, il appartient toujours aux États de prendre les décisions finales et d'assumer les responsabilités juridiques dans le cadre d'une procédure d'autorisation.

Bien que les normes de sûreté établissent une base essentielle pour la sûreté, il est aussi parfois nécessaire d'incorporer des prescriptions plus détaillées conformément à l'usage national. De surcroît, il y aura souvent des aspects particuliers qui devront être soumis, cas par cas, à l'appréciation de spécialistes.

La protection physique des produits fissiles et des matières radioactives, comme celle de la centrale nucléaire dans son ensemble, est mentionnée là où il convient, mais n'est pas traitée en détail ; pour connaître les obligations des États à cet égard, il convient de se reporter aux instruments et aux publications pertinents élaborés sous les auspices de l'AIEA. Les aspects non radiologiques de la sécurité du travail et de la protection de l'environnement ne sont pas non plus explicitement examinés ; il est admis que les États devraient se conformer aux obligations et aux engagements internationaux qu'ils ont contractés dans ce domaine.

Les prescriptions et recommandations présentées dans les normes de sûreté de l'AIEA peuvent n'être pas pleinement satisfaites par certaines installations anciennes. Il appartient à chaque État de statuer sur la manière dont les normes seront appliquées à ces installations.

Il convient d'attirer l'attention des États sur le fait que les normes de sûreté de l'AIEA, bien que n'étant pas juridiquement contraignantes, visent à faire en sorte que l'énergie nucléaire et les matières radioactives utilisées à des fins pacifiques le soient d'une manière qui permette aux États de s'acquitter des obligations qui leur incombent en vertu des principes du droit international et de règles recueillant l'assentiment général, tels que ceux qui concernent la protection de l'environnement. En vertu de l'un de ces principes, le territoire d'un État ne doit pas servir à des activités qui portent préjudice à un autre État. Les États sont donc tenus de faire preuve de prudence et d'observer des normes de conduite.

Comme toute autre activité, les activités nucléaires civiles menées sous la juridiction des États sont soumises aux obligations que les États contractent au titre de conventions internationales, en sus des principes du droit international généralement acceptés. Les États sont censés adopter au niveau national les lois (et la réglementation), ainsi que les normes et mesures dont ils peuvent avoir besoin pour s'acquitter efficacement de toutes leurs obligations internationales.

NOTE DE L'ÉDITEUR

Lorsqu'une norme comporte un appendice, ce dernier est réputé faire partie intégrante de cette norme et avoir le même statut que celle-ci. En revanche, les annexes, notes infrapaginales et bibliographies ont pour objet de donner des précisions ou des exemples concrets qui peuvent être utiles au lecteur.

Le présent a été employé pour énoncer des prescriptions, des responsabilités et des obligations. Le conditionnel sert à énoncer des recommandations concernant une option souhaitable.

La version anglaise du texte est celle qui fait autorité. La présente traduction a été établie sous les auspices de l'Institut de radioprotection et de sûreté nucléaire (IRSN) (France).

TABLE DES MATIÈRES

1. INTRODUCTION 1

Généralités (1.1–1.4) 1
Objectif (1.5) 2
Champ d'application (1.6–1.10) 2
Structure (1.11–1.14) 3

2. CONSIDÉRATIONS TECHNIQUES RELATIVES AUX SYSTÈMES PROGRAMMÉS 3

Caractéristiques des systèmes programmés (2.1–2.3) 3
Processus de développement (2.4–2.8) 4
Problèmes de sûreté et de fiabilité (2.9–2.11) 7
Problèmes organisationnels et juridiques (2.12–2.16) 8

3. APPLICATION DES EXIGENCES RELATIVES À LA GESTION DE LA SÛRETÉ AUX SYSTÈMES PROGRAMMÉS (3.1) 9

Exigences relatives à la gestion de la sûreté (3.2–3.20) 9
Activités de conception et de développement (3.21–3.27) 14
Gestion et assurance de la qualité (3.28–3.33) 15
Documentation (3.34–3.44) 17

4. PLANIFICATION DU PROJET (4.1–4.2) 19

Plan de développement (4.3-4.10) 20
Assurance de la qualité (4.11) 21
Plan de vérification et de validation (4.12–4.18) 22
Plan de gestion de la configuration (4.19–4.24) 24
Plan d'installation et de mise en service (4.25–4.26) 25

5. EXIGENCES RELATIVES AU SYSTÈME PROGRAMMÉ (5.1–5.3) . . . 26

Recommandations (5.4–5.23) 27
Documents (5.24–5.40) 31

6. CONCEPTION DU SYSTÈME PROGRAMMÉ (6.1–6.2) 34

Recommandations (6.3–6.26) 34
Documents (6.27–6.42) 39

7. EXIGENCES RELATIVES AU LOGICIEL (7.1–7.4) 43

Recommandations (7.5–7.17) 44
Documents (7.18–7.21) 47

8. CONCEPTION DU LOGICIEL (8.1–8.3) 48

Recommandations (8.4–8.12) 49
Documents (8.13–8.23) 50

9. IMPLÉMENTATION DU LOGICIEL (9.1–9.2) 52

Recommandations (9.3–9.27) 53
Documents (9.28–9.32) 58

10. VÉRIFICATION ET ANALYSE (10.1) 58

Recommandations (10.2–10.33) 59
Documents (10.34–10.41) 65

11. INTÉGRATION DU SYSTÈME PROGRAMMÉ (11.1–11.2) 66

Recommandations (11.3–11.13) 66
Documents (11.14–11.15) 68

12. VALIDATION DES SYSTÈMES PROGRAMMÉS (12.1–12.2) 69

Recommandations (12.3–12.15) 69
Documents (12.16) 71

13. INSTALLATION ET MISE EN SERVICE (13.1–13.4) 72

Recommandations (13.5–13.10) 73
Documents (13.11) 74

14. EXPLOITATION (14.1–14.2) 74

Recommandations (14.3–14.9) 75
Documents (14.10–14.12) 76

15. MODIFICATIONS APRÈS LIVRAISON (15.1) 77

Recommandations (15.2–15.8) 77
Documents (15.9–15.12) 78

RÉFÉRENCES 81

ANNEXE : UTILISATION ET VALIDATION D'UN LOGICIEL PRÉEXISTANT 83

GLOSSAIRE 89
PERSONNES AYANT COLLABORÉ À LA RÉDACTION ET À L'EXAMEN 93
ORGANISMES CONSULTATIFS POUR L'APPROBATION DES NORMES DE SÛRETÉ 95

1. INTRODUCTION

GÉNÉRALITÉS

1.1. Les systèmes programmés prennent de plus en plus d'importance pour la sûreté des centrales nucléaires étant donné que leur utilisation, tant dans les nouvelles centrales que dans les centrales plus anciennes, s'accroît rapidement. Ils sont utilisés pour les applications liées à la sûreté, comme certaines fonctions des systèmes de surveillance et de contrôle de processus, ainsi que pour les applications critiques de sûreté, comme la protection du réacteur ou la mise en service des systèmes de sûreté. Il est donc de première importance, vis-à-vis de la sûreté, d'assurer la sûreté de fonctionnement des systèmes informatisés.

1.2. Avec la technologie actuelle, il est en principe possible de développer des systèmes programmés d'instrumentation et de contrôle-commande, pour les systèmes importants pour la sûreté, qui possèdent la capacité d'améliorer le niveau de sûreté et de fiabilité avec une sûreté de fonctionnement suffisante. Cependant, leur sûreté de fonctionnement ne peut être prévue et prouvée que si un processus d'ingénierie systématique, entièrement documenté et analysable, est respecté. Bien qu'un certain nombre de normes nationales et internationales, se rapportant à l'assurance de la qualité des systèmes programmés importants pour la sûreté, aient été préparées, il n'est généralement pas possible de disposer de critères mondialement approuvés prouvant la sûreté de tels systèmes. Il est évident qu'il peut exister d'autres moyens, différents de ceux recommandés ici, pour apporter la preuve que le niveau de sûreté nécessaire est atteint.

1.3. Les exigences fondamentales relatives à la conception de systèmes de sûreté destinés aux centrales nucléaires sont indiquées dans les prescriptions de sûreté relatives à la conception publiées dans la collection Normes de sûreté de l'AIEA [1]. Ces exigences ont été étendues et adaptées à la conception du système de protection dans la réf. [2] et à la conception des systèmes d'instrumentation et de contrôle dans la réf. [3]. La révision des réf. [2, 3] est en cours afin de refléter le niveau actuel de la technologie, y compris l'application de la technologie numérique.

1.4. L'AIEA a publié un rapport technique [4] pour aider les États Membres à garantir que les systèmes programmés importants pour la sûreté des centrales nucléaires sont sûrs et que les autorisations sont correctement délivrées. Le rapport fournit les informations sur les règles de l'art en matière de logiciels qui, accompagnées des normes correspondantes (telles que réf. [5]), constituent une base technique pour ce Guide de sûreté.

OBJECTIF

1.5. L'objectif de ce Guide de sûreté est de fournir des conseils sur la collecte des éléments de justification et la préparation des documents à utiliser pour prouver la sûreté du logiciel des systèmes programmés importants pour la sûreté des centrales nucléaires, dans toutes les phases du cycle de vie du système.

CHAMP D'APPLICATION

1.6. Ces conseils s'appliquent aux systèmes importants pour la sûreté tels que définis dans les réf. [2, 3]. Étant donné qu'actuellement la fiabilité d'un système programmé ne peut être prévue en se basant uniquement sur le processus de conception, ou intégrée par ce dernier, il est difficile de définir et de convenir de manière systématique des assouplissements vis-à-vis des conseils à appliquer au logiciel destiné aux systèmes liés à la sûreté. À chaque fois que cela est possible, les recommandations qui ne s'appliquent qu'aux systèmes de sûreté et non pas aux systèmes liés à la sûreté sont explicitement identifiées.

1.7. Les conseils s'adressent principalement au logiciel utilisé dans les systèmes programmés importants pour la sûreté. Les conseils relatifs aux autres aspects des systèmes programmés, tels que ceux qui concernent la conception du système programmé lui-même et du matériel correspondant, sont limités aux problèmes soulevés par le développement, la vérification et la validation du logiciel.

1.8. Ce Guide de sûreté se concentre principalement sur la préparation des documents utilisés pour démontrer de manière adéquate la sûreté et la fiabilité des systèmes programmés importants pour la sûreté.

1.9. Ce Guide de sûreté s'applique à tout type de logiciel : micrologiciel ou logiciel préexistant (tel qu'un système d'exploitation), logiciel à développer spécifiquement pour le projet ou logiciel à développer à partir d'une gamme d'équipement prédéveloppé existante de matériel informatique ou de modules logiciels. Le problème de l'utilisation pour des fonctions de sûreté d'un logiciel préexistant ou d'un logiciel de série du commerce, sur le développement desquels on dispose de peu d'informations, est traité en annexe, où une section de la réf. [6] est reproduite (voir également la section 6.3 de la réf. [7]). Les informations concernant la spécification de prescriptions au cours de mises à niveau et de systèmes d'instrumentation et de contrôle-commande peuvent être consultées dans la réf. [8].

1.10. Ce Guide de sûreté est destiné aux personnes impliquées dans la production, l'évaluation et la délivrance des autorisations de systèmes programmés, dont les

concepteurs de systèmes de réacteurs, les concepteurs et programmeurs de logiciels, les vérificateurs, les responsables de la validation, les responsables de la certification, les responsables de la réglementation, ainsi que les opérateurs de réacteurs nucléaires. Les diverses interfaces entre ceux qui sont concernés sont étudiées.

STRUCTURE

1.11. La section 2 fournit des recommandations relatives aux aspects techniques des systèmes programmés, traitant des avantages et des inconvénients de tels systèmes, des problèmes de sûreté et de fiabilité, et de certaines conditions préalables d'organisation du projet de développement du système.

1.12. La section 3 fournit des recommandations sur l'application des prescriptions pour la gestion de la sûreté aux systèmes programmés importants pour la sûreté.

1.13. La section 4 fournit des recommandations sur la phase de planification du projet de développement du système et décrit la structure et le contenu des documents associés, dont le plan de développement, la description du programme d'assurance de la qualité, le plan de vérification et de validation et le plan de gestion de la configuration.

1.14. Les sections 5 à 15 sont réservées aux phases individuelles du cycle de vie du développement. Les sections commencent par une brève introduction qui décrit chaque phase. Sous la rubrique RECOMMANDATIONS se trouvent un ensemble de recommandations pour cette phase. Sous la rubrique DOCUMENTS se trouve la liste des documents à produire comme résultats à la fin de la phase, et des conseils concernant le contenu de ces documents sont fournis. De plus, des recommandations d'ordre général sont formulées en ce qui concerne les attributs et la présentation des produits de la phase. Dans toutes les parties, l'intention n'est pas de fournir une description exhaustive de tous les documents qui seront nécessaires au développement, mais plutôt de récapituler les recommandations et les documents, accompagnés de leurs attributs, les plus importants pour démontrer la sûreté du système.

2. CONSIDÉRATIONS TECHNIQUES RELATIVES AUX SYSTÈMES PROGRAMMÉS

CARACTÉRISTIQUES DES SYSTÈMES PROGRAMMÉS

2.1. Vis-à-vis de l'évaluation de la sûreté et de la fiabilité, les systèmes programmés possèdent deux propriétés fondamentales. Ils sont programmables et le matériel est

basé sur une logique numérique discrète. Comme pour les autres systèmes, les défaillances du matériel peuvent être dues à des erreurs de conception ou de fabrication, mais généralement elles sont le résultat d'une usure, de processus de dégradation ou environnementaux et sont de nature aléatoire. Le logiciel, la partie programmable de l'ordinateur, ne subit pas d'usure mais peut être affecté par des modifications de l'environnement d'exploitation. Les défaillances logicielles peuvent résulter de spécifications d'exigences erronées ou manquant de clarté (ce qui entraîne des erreurs de conception logique ou d'implémentation) ou d'erreurs lors de phase d'implémentation ou de maintenance.

2.2. La nature programmable des systèmes programmés, conjuguée à la logique discrète, implique qu'ils possèdent un certains nombre d'avantages par rapport aux systèmes non programmables et non numériques. Ils facilitent la réalisation de fonctions complexes ; ils peuvent, en particulier, fournir une surveillance améliorée des paramètres de la centrale, des interfaces opérateur améliorées, des équipements de diagnostic, d'autocontrôle, d'étalonnage et de tests améliorés. Ils peuvent également offrir une précision et une stabilité plus grandes. L'utilisation de structures de type 'bus' multiplexées peuvent réduire les volumes de câblage. Les modifications du logiciel requièrent moins de démontage de l'équipement, ce qui peut être utile pour la maintenance.

2.3. Ces avantages sont contrebalancés par un certain nombre d'inconvénients. L'implémentation du logiciel tend à devenir plus complexe et, de ce fait, plus sujette à des erreurs de conception que l'implémentation de systèmes uniquement câblés. De plus, les implémentations de logiciels sont des modèles en logique discrète du monde réel. Ceci amène deux types de conséquences. Le logiciel est plus sensible (i.e. moins tolérant) vis-à-vis des 'petites' erreurs. Il est également plus difficile à tester, parce que l'interpolation et l'extrapolation sont beaucoup plus difficiles à appliquer aux systèmes programmés qu'aux systèmes analogiques traditionnels, et en fin de compte ne sont pas complètement valides.

PROCESSUS DE DÉVELOPPEMENT

2.4. Le développement de systèmes importants pour la sûreté devrait être un processus contrôlé étape par étape. Dans cette optique, le processus de développement est organisé sous forme d'une ensemble ordonné de phases distinctes. Chaque phase utilise les informations obtenues lors des phases précédentes et fournit les informations d'entrée pour les phases suivantes. Le développement de systèmes importants pour la sûreté est par nature un processus itératif. Au fur et à mesure de la progression de la conception, les défauts et omissions intervenus lors des étapes antérieures

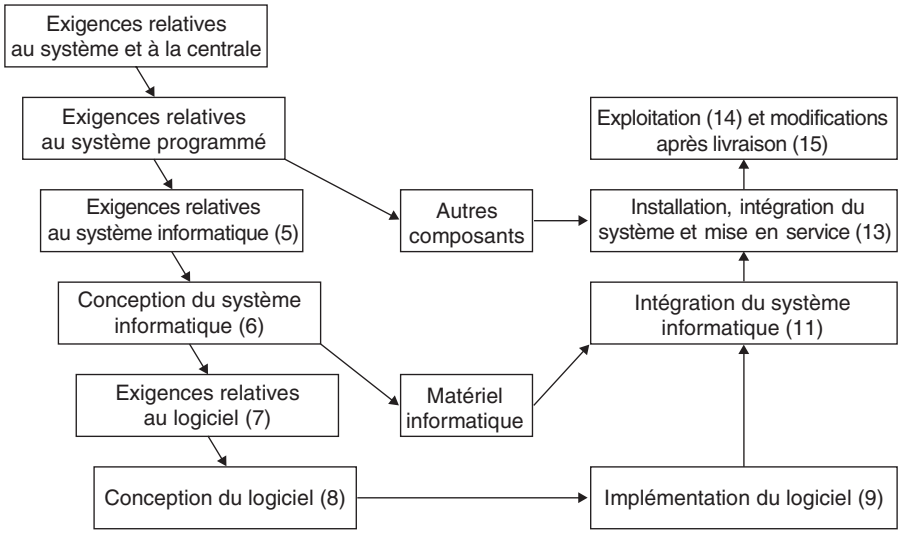

FIG. 1. Développement d'un logiciel de systèmes programmés importants pour la sûreté (les numéros font référence aux sections de ce Guide de sûreté).

apparaissent et nécessitent des itérations sur les étapes précédentes. Une caractéristique essentielle de cette méthode est que les produits de chaque phase de développement sont vérifiés par rapport aux exigences de la phase précédente. À certaines phases du développement, un processus de validation est exécuté afin de confirmer la conformité du produit par rapport aux exigences fonctionnelles et non fonctionnelles ainsi que l'absence de comportement non désiré. Les avantages principaux d'un tel processus contrôlé étape par étape sont décrits dans la section 3.2.2 de la réf. [4].

2.5. Les phases typiques du processus de développement et un bref exposé du processus pouvant être appliqué sont présentés sur la figure 1. Les cadres présentent les activités de développement à effectuer et les flèches indiquent l'ordre prévu et le flux d'informations principales. Les numéros entre parenthèses sur la figure 1 indiquent les sections de ce Guide de sûreté dans lesquelles sont décrits les produits et les activités du développement. Les activités sans référence ne font pas partie du champ d'application de ce Guide de sûreté mais sont présentées pour préciser le contexte. La figure 2 illustre les relations de la vérification et de la validation avec les exigences, la conception et l'implémentation. Le choix des activités de développement spécifiques et leur ordre sur ces figures et dans le Guide de sûreté ne sont pas destinés à imposer une méthode spécifique de développement ; des variantes peuvent posséder les attributs nécessaires et être capables de satisfaire aux exigences.

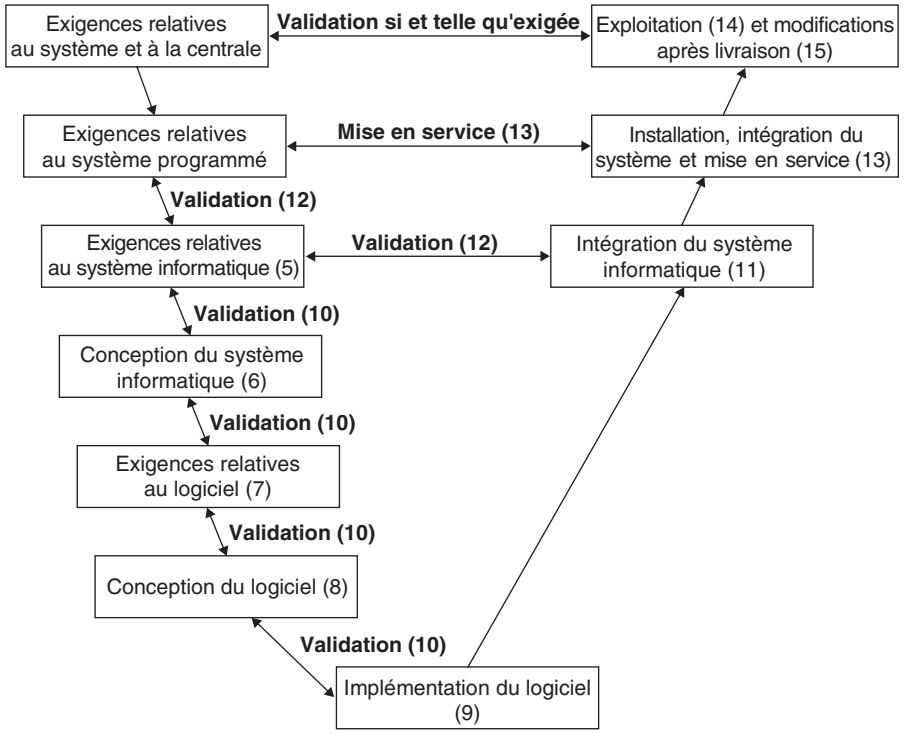

FIG. 2. Vérification, validation et mise en service (les numéros font référence aux sections de ce Guide de sûreté).

2.6. Le développement du système programmé devrait commencer par une analyse des exigences de la centrale et des systèmes exécutée par des spécialistes en ingénierie, incluant des ingénieurs de sûreté et des ingénieurs en logiciel. En fonction des résultats de cette analyse, on déduit les exigences relatives au système programmé. Normalement, des itérations sont nécessaires dans cette analyse avant de pouvoir finaliser un ensemble d'exigences. Le caractère systématique de cette méthode déductive devrait être clairement prouvé car les omissions à ce stade peuvent entraîner une spécification incorrecte des exigences concernant la sûreté et ainsi entraîner un système non sûr.

2.7. La première décision de conception devrait être de répartir les exigences concernant le système informatisé entre le système programmé (prescriptions relatives au système programmé) et l'équipement électronique et électrique conventionnel servant à la mesure des paramètres de la centrale et à la mise en service des appareils de contrôle-commande (autres composants). Le concepteur peut avoir des raisons

d'opter pour un équipement analogique afin de mettre en œuvre certaines exigences fonctionnelles[1].

2.8. Les exigences relatives au système programmé sont réparties entre les exigences logicielles et le matériel informatique pour la conception du système informatique. Le logiciel est alors conçu sous forme d'un ensemble de modules interagissant entre eux, implémentés sous forme de programme qui s'exécutera sur un ordinateur (conception du logiciel et implémentation du logiciel). Ensuite, le logiciel est intégré dans l'ordinateur afin de créer le système programmé (intégration du système programmé). Finalement, le système programmé est installé (installation) dans la centrale afin de procéder à la mise en service et au fonctionnement. Une des étapes de la phase de mise en service est l'intégration du système programmé aux autres composants (ce qui fait partie de la phase d'intégration du système).

PROBLÈMES DE SÛRETÉ ET DE FIABILITÉ

2.9. À cause des inconvénients mentionnés précédemment, l'évaluation quantitative de la fiabilité d'un système basé sur un logiciel est plus difficile que pour les systèmes non programmables et ceci peut entraîner des difficultés lors de la démonstration de la sûreté attendue d'un système programmé. On ne peut pas actuellement démontrer les allégations relatives à la haute fiabilité d'un logiciel. En conséquence, les conceptions qui requièrent une probabilité de défaillance à la demande d'un système programmé inférieure à 10^{-4} doivent être considérées avec prudence (voir par. 8.2.2 de la réf. [9]).

2.10. Étant donné que les défaillances des logiciels sont, par nature, systématiques plutôt qu'aléatoires, les défaillances de cause commune des systèmes programmés de sûreté employant des sous-systèmes redondants utilisant des copies identiques du logiciel sont un problème de première importance. Les contre-mesures ne sont pas simples à mettre en œuvre. Les concepteurs supposent l'indépendance et utilisent la diversification et une stratégie de qualification exhaustive pour se prémunir contre les défaillances de cause commune. Cependant, il peut être difficile d'estimer le degré de réussite et les avantages de ces stratégies lorsque qu'un logiciel est impliqué.

2.11. Grâce à la technologie actuelle, il est en principe possible de développer des systèmes programmés d'instrumentation et de contrôle-commande possédant une

[1] Ce Guide de sûreté ne traite pas des aspects de l'évaluation se rapportant à d'autres équipements ; il se concentre sur le système programmé et plus particulièrement sur le logiciel s'exécutant sur l'ordinateur.

sûreté de fonctionnement nécessaire pour des systèmes importants pour la sûreté, et de prouver qu'ils sont suffisamment sûrs. Toutefois, la sûreté de fonctionnement ne peut être prouvée que si une procédure méticuleuse et complètement documentée est respectée. Cette procédure peut inclure l'évaluation de l'expérience d'exploitation avec un logiciel préexistant, suivant des exigences spécifiques (voir également l'annexe). Les recommandations permettant de réaliser une démonstration adéquate de sûreté sont fournies dans ce Guide de sûreté.

PROBLÈMES ORGANISATIONNELS ET JURIDIQUES

2.12. Il existe divers aspects organisationnels et juridiques se rapportant au projet de développement d'un système programmé qui devraient être traités en profondeur au tout début du projet afin d'assurer sa réussite. Ils comportent des facteurs tels que l'existence d'un contexte administratif et juridique adéquat pour l'autorisation réglementaire de mise en service des systèmes programmés importants pour la sûreté et de ressources et compétences suffisantes au sein des organisations impliquées dans le processus de développement des systèmes. Si ces facteurs ne sont pas soigneusement étudiés au stade de la conception du projet, le planning et les coûts de ce dernier peuvent en être considérablement affectés. Certains de ces facteurs peuvent influencer la décision d'utiliser une technologie numérique programmable, rendant cette option impraticable ou même la condamnant à cause de son prix de revient.

2.13. La quantification de la fiabilité du logiciel est un problème non résolu. Le test du logiciel possède ses propres limites et la quantification de la fiabilité du logiciel pour les systèmes programmés peut être difficile ou impossible à faire. Les prises de position réglementaires vis-à-vis de la démonstration de la sûreté et de la fiabilité requise pour le logiciel devraient être clarifiées très tôt dans l'étape de planification du projet. La méthode à suivre pour traiter les problèmes de sûreté et de fiabilité devrait être déterminée, documentée, mise à la disposition de, et si nécessaire agréée par, l'organisme de réglementation. Ceci peut inclure des points d'arrêt spécifiques réglementaires.

2.14. On devrait s'assurer que des ressources et compétences suffisantes sont mises à la disposition de l'organisme de réglementation, de l'équipe de conception de l'exploitant, de l'équipe d'assistance technique de l'organisme de réglementation et du fournisseur pour satisfaire aux recommandations (par exemple celles de ce Guide de sûreté) relatives au processus de développement du logiciel et à la démonstration de sa sûreté. L'exploitant devrait également s'assurer que le fournisseur de l'ordinateur et/ou du logiciel sera prêt à mettre à disposition toutes les informations qu'il possède nécessaires au processus de délivrance de l'autorisation.

2.15. L'exploitant (i.e. l'utilisateur du système programmé) devrait créer une structure appropriée pour le traitement des problèmes d'exploitation et de maintenance. Il est possible que l'exploitant ait besoin de sa propre équipe pour effectuer des modifications sur le logiciel après sa livraison. Cette équipe doit posséder le même niveau de compétence et d'équipement que celui du constructeur d'origine.

2.16. Ces facteurs contribuent à une augmentation significative des ressources techniques nécessaires et, de ce fait, ont une incidence sur les coûts. Les implications financières peuvent être telles que l'option d'un système informatisé peut se révéler être d'un prix prohibitif.

3. APPLICATION DES EXIGENCES RELATIVES À LA GESTION DE LA SÛRETÉ AUX SYSTÈMES PROGRAMMÉS

3.1. La majorité des exigences de la sûreté développées dans le passé pour les systèmes non programmés sont applicables aux systèmes programmés. Toutefois, l'application de ces exigences concernant la sûreté aux systèmes programmés, vu les différences entre le logiciel et le matériel, n'est pas toujours évidente et possède, souvent, de nouvelles implications. La section 3 fournit un bref aperçu des exigences de sûreté relatives à la conception, au développement et à l'assurance de la qualité ainsi qu'à la documentation. Ces exigences relatives à la sûreté constituent la base de l'élaboration des exigences relatives au système programmé, spécialement pour les exigences non fonctionnelles. La documentation devrait être de haute qualité, vu son importance pour la conception du logiciel et pour la démonstration de la sûreté.

EXIGENCES RELATIVES À LA GESTION DE LA SÛRETÉ

Simplicité de conception

3.2. Il devrait être démontré que toute complexité inutile a été évitée dans la fonctionnalité du système ainsi que dans son implémentation. Cette démonstration est importante pour la sûreté et n'est pas évidente, car la technologie programmable numérique permet la réalisation de fonctionnalité plus complexe. La preuve du respect d'une conception structurée, d'une discipline de programmation et de règles de codification devrait faire partie de cette démonstration.

3.3. Pour les systèmes de sûreté, les exigences fonctionnelles auxquelles devrait satisfaire un système programmé doivent toutes être indispensables dans l'accomplissement de fonctions de sûreté, les fonctions non indispensables pour la sûreté devraient en être séparées et devraient être présentées comme n'influant pas sur les fonctions de sûreté.

3.4. Pour les applications de systèmes programmés, la décomposition descendante, les niveaux de séparation et la structure modulaire sont des concepts importants pour réagir face aux problèmes d'une complexité inévitable. Cela permet non seulement aux développeurs système de s'attaquer à des problèmes plus petits et plus gérables mais permet également au vérificateur de procéder à un examen plus efficace. La logique qui soutient la modularisation du système et la définition des interfaces devraient être aussi simple que possible (par exemple en appliquant le « masquage d'informations » (voir section 3.3.4 de la réf. [4]).

3.5. Lors de la conception des modules du système, il faudrait préférer les algorithmes simples à ceux qui sont complexes. La simplicité ne devrait pas être sacrifiée pour atteindre des performances non requises. L'ordinateur utilisé dans les systèmes de sûreté devrait être spécifié avec une capacité et des performances suffisantes de façon à empêcher le logiciel de devenir trop compliqué.

Culture de sûreté

3.6. Le personnel travaillant sur un projet de développement de logiciel destiné à un système très important pour la sûreté devrait inclure des spécialistes de l'application projetée ainsi que des spécialistes en logiciel et en matériel. Cette combinaison d'expertise aide à s'assurer que les exigences relatives à la sûreté, qui sont généralement bien connues en raison de la maturité de ce secteur industriel, sont communiquées efficacement aux spécialistes informatiques. Il faudrait s'assurer que la totalité du personnel comprend la relation entre leurs travaux et l'accomplissement des exigences relatives à la sûreté et ce personnel devrait être encouragé à remettre en question les activités ou les décisions pouvant compromettre la sûreté du système. Cela signifie que les spécialistes en logiciel devraient posséder également une bonne compréhension de l'application projetée. Un langage de spécification approprié (par exemple un langage graphique) peut être utilisé pour la description des fonctions de sûreté.

Système de classification de la sûreté

3.7. Un système de classification de la sûreté permettant de définir l'importance pour la sûreté des fonctions d'instrumentation et de contrôle-commande peut être

utilisé pour le processus de développement du système. Il conditionne le degré d'attention approprié (voir par. 3.8) porté par les concepteurs, les exploitants et les autorités réglementaires aux spécifications, à la conception, à l'assurance de la qualité, à la fabrication, à l'installation, à la maintenance et aux essais du système et des équipements, qui permet d'assurer la sûreté.

Équilibre entre la réduction des risques et l'effort de développement

3.8. Les compromis relatifs à la réalisation de divers objectifs de conception contradictoires devraient être soigneusement évalués à chaque étape de la conception du système et de son logiciel associé. Un processus de développement et de conception descendant devrait être utilisé pour faciliter cette évaluation. Des exigences de qualification et de conception graduées, lorsqu'elles s'appliquent à des fonctions du système programmé, peuvent découler d'un système de classification de la sûreté (par. 3.7). Cette graduation peut être utilisée pour équilibrer l'effort de qualification et de conception. Le système programmé devrait satisfaire aux critères de la classe de sûreté la plus élevée des fonctions qu'il implémente.

3.9. Des mesures appropriées garantissant le niveau de confiance nécessaire devraient être associées à chaque classe de sûreté. Il doit être noté que, pour la partie matérielle du système, le niveau de confiance peut être évalué à l'aide de techniques quantitatives ; cependant, pour le logiciel, seule une évaluation qualitative est possible.

Défense en profondeur

3.10. La défense en profondeur telle qu'appliquée à la conception des réacteurs nucléaires [1, 10] devrait être utilisée dans le développement du système programmé et de son logiciel associé. Si un système programmé constitue une fonction de sûreté principale, la défense en profondeur, par exemple au moyen d'un système diversifié de protection secondaire, devrait être appliquée.

Redondance

3.11. La conception traditionnelle basée sur des chaînes d'instrumentation à redondance multiple avec un vote logique, telle qu'utilisée pour les applications analogiques, s'applique avantageusement au matériel de systèmes programmés. Toutefois, ce type de redondance n'évite pas les pannes du système dues aux défauts d'origine commune de conception du matériel et du logiciel pouvant conduire à des défaillances de tous les circuits redondants.

Critère de défaillance unique

3.12. L'application du critère de défaillance unique aux défaillances aléatoires du matériel est directe [1, 11] : aucune défaillance unique ne doit entraîner de perte des fonctions de sûreté. Cependant, pour le logiciel, ce critère est difficile à satisfaire, étant donné qu'un défaut causant une panne du logiciel est nécessairement présent sur toutes les répliques de ce logiciel (voir par. 3.13).

Diversité

3.13. La fiabilité des systèmes programmés peut être améliorée en utilisant la diversité pour réduire le potentiel de défaillances d'origine commune du logiciel. L'utilisation de fonctions et de composants système diversifiés à différents niveaux de la conception devrait être envisagée. La diversité des méthodes, langages, outils et du personnel devrait également être prise en compte. Il faut noter cependant que, bien qu'un logiciel différent puisse améliorer la protection contre les défaillances de cause commune de logiciel, il ne garantit pas l'absence d'erreurs coïncidentes. La décision de l'exploitant d'utiliser la diversité, le choix du type de diversité ou la décision de ne pas utiliser la diversité devraient être justifiés.

Système à pannes sûres, surveillance et tolérance aux défaillances

3.14. Des systèmes à pannes sûres et des moyens de surveillance et de tolérance aux défaillances devraient être intégrés au logiciel, mais seulement dans la mesure où la complexité supplémentaire est justifiée par une augmentation globale démontrable de la sûreté. Lorsque le logiciel est utilisé pour vérifier le matériel sur lequel il tourne, alors sa capacité à répondre correctement devrait également être prouvée. L'utilisation de dispositifs externes, tels que des dispositifs « chiens de garde », rend la réponse du système à la détection de défaillance plus fiable. Une programmation défensive, des langages appropriés, incluant des sous-ensembles sécurisés de langage, devraient être employés pour assurer une réponse sûre en toutes circonstances, dans la mesure du possible. Un des objectifs de la phase de définition des exigences du système programmé devrait être de spécifier complètement la réponse sûre et désirée à toutes les combinaisons d'entrées.

Sécurité

3.15. Il devrait être démontré que des mesures ont été prises pour protéger le système programmé durant toute sa vie contre des attaques physiques, des intrusions intentionnelles et non intentionnelles, fraudes, virus, etc. [12, 13]. Les systèmes de sûreté

ne devraient pas être connectés à des réseaux externes lorsque la sûreté d'un tel acte ne peut pas être justifiée.

Maintenabilité

3.16. Le système programmé devrait être conçu de manière à faciliter la détection, la localisation et le diagnostic des pannes pour pouvoir efficacement réparer ou remplacer le système. Un logiciel possédant une structure modulaire sera plus facile à réparer, et plus facile également à examiner et analyser étant donné que la conception peut être plus simple à appréhender et à modifier sans introduire de nouvelles erreurs. La maintenabilité du logiciel inclut également le concept d'apport de modifications à la fonctionnalité. La conception d'un système programmé devrait permettre d'assurer que les modifications sont confinées sur une petite partie du logiciel.

Représentation exhaustive des modes de fonctionnement

3.17. Les exigences et la conception du logiciel des systèmes importants pour la sûreté devraient définir explicitement toutes les relations entre l'entrée et la sortie pour chacun des modes de fonctionnement. La conception du logiciel devrait être suffisamment simple pour permettre l'examen de toutes les combinaisons d'entrée que présentent tous les modes de fonctionnement.

Interfaces homme–machine et prise en compte des limites humaines

3.18. La conception des interfaces homme-machine peut avoir un impact significatif sur la sûreté. Les interfaces homme–machine devraient être conçues de manière à fournir à l'opérateur une somme d'informations suffisante et structurée mais non écrasante, et à lui fournir également suffisamment de temps pour réagir (voir, par exemple, la règle des trente minutes dans la réf. [2]). Lorsque des fonctions sont attribuées à l'opérateur, le système programmé devrait accorder le temps nécessaire pour définir une action manuelle à partir des informations fournies. La validité de toutes les données d'entrée devrait être vérifiée pour se prémunir de défaillances dues à une erreur de l'opérateur. La possibilité pour l'opérateur de déroger à ces contrôles de validité devrait être soigneusement étudiée.

Sûreté de fonctionnement démontrable

3.19. Il ne suffit pas que le système soit fiable, il faut également pouvoir démontrer au responsable de la réglementation qu'il est fiable. Ce Guide de sûreté fournit des recommandations aux exploitants quant à la façon d'atteindre une sûreté de fonctionnement démontrable par le biais des méthodes de conception et de qualification qui

améliorent la traçabilité ainsi que par le biais de la production des documents adéquats.

Testabilité

3.20. Chaque exigence et chaque caractéristique de conception devraient être exprimées de telle manière qu'un test puisse être effectué pour déterminer si cette caractéristique a été correctement implémentée. Les exigences fonctionnelles et non fonctionnelles devraient pouvoir être testées. Les résultats de test devraient posséder une traçabilité permettant de remonter aux exigences associées.

ACTIVITÉS DE CONCEPTION ET DE DÉVELOPPEMENT

3.21. Lors de la détermination des activités nécessaires de conception et de développement, une attention particulière devrait être donnée aux sujets suivants (par. 3.22–3.27).

Processus contrôlé étape par étape

3.22. Le processus de conception et de développement devrait être contrôlé étape par étape. Ce processus de développement peut témoigner de son exactitude par sa construction même. Ceci peut également faciliter le processus de vérification et assurer la détection précoce des erreurs dans le processus de conception.

Scrutabilité

3.23. Des documents précis et facilement scrutables devraient être fournis à toutes les étapes du processus de développement. Les documents utilisés pour démontrer l'adéquation au responsable de la réglementation devraient être identiques à ceux réellement utilisés dans la conception.

Essais complets

3.24. Une batterie complète de tests devrait être appliquée. Les essais sont une partie importante du développement, de la vérification et de la validation. La démonstration de la couverture des tests, comprenant la traçabilité de la liaison entre les essais et les documents source, devrait être fournie. Les résultats de tests, la preuve de la couverture de tests et autres enregistrements de tests devraient être disponibles en vue d'un audit effectué par une tierce partie. Un plan d'essais complet devrait être élaboré et mis à la disposition du responsable de la réglementation au tout début du projet.

Utilisation d'outils automatisés

3.25. Tous les outils utilisés devraient être compatibles entre eux. Les outils devraient être qualifiés à un niveau proportionné à leur fonction dans le développement du logiciel et dans la démonstration de la sûreté. Les techniques utilisées pour acquérir un niveau de confiance dans les outils devraient être spécifiées et documentées.

Traçabilité

3.26. Les exigences devraient être traçables jusqu'à la conception ; la conception devrait être traçable jusqu'à la programmation ; les exigences, conception et programmation devraient conduire aux essais. La traçabilité devrait être conservée lorsque des modifications sont effectuées. Il devrait également exister une traçabilité dans l'autre sens afin de s'assurer qu'aucune fonction non désirée a été créée.

Conformité aux normes

3.27. Les exigences relatives à la gestion de la sûreté, les exigences de sûreté et les normes techniques devant être utilisées dans le développement devraient être identifiées. Une analyse de conformité devrait être préparée pour la principale norme [5] utilisée dans la spécification et la réalisation des systèmes programmés.

GESTION ET ASSURANCE DE LA QUALITÉ

Fonctions et qualifications du personnel clairement définies

3.28. La direction de la centrale devrait démontrer que le niveau de la dotation en personnel est adéquat. La direction devrait établir les fonctions et qualifications requises pour le personnel concerné par la conception du logiciel, les programmes de production et de maintenance, et devrait garantir que seul le personnel qualifié et expérimenté exécute ces fonctions. Les qualifications du personnel devraient être établies pour chacune des tâches au sein du processus de développement et de maintenance du logiciel, y compris l'exécution du programme d'assurance de la qualité [14].

Pratiques acceptables

3.29. Des méthodes, langages et outils bien établis devraient être utilisés pour le développement du logiciel. On ne devrait pas utiliser des méthodes, langages et outils qui en sont encore au stade de recherche.

Programme d'assurance de la qualité

3.30. Le programme d'assurance de la qualité de l'organisation devrait s'étendre au processus de développement du logiciel, et devrait également couvrir la gestion de la configuration et le contrôle des modifications après livraison du logiciel. Pour les systèmes de sûreté au moins, les tests, validation et vérification indépendants, avec des audits auxiliaires indépendants, devraient également être couverts. Les exigences de qualité pour le logiciel devraient être décrites dans un plan d'assurance qualité du logiciel qui peut être exigé par le programme d'assurance qualité.

Attribution des responsabilités

3.31. Des interfaces devraient être créées entre les entités organisationnelles au sein de l'organisation chargée de la conception et entre l'organisation chargée de la conception, son client et les autres organisations impliquées dans le processus de développement du système. Les contrôles se rapportant à l'interfaçage de la conception devraient être établis et devraient inclure l'attribution des responsabilités et la distribution des documents provenant des organisations en interface ou leur étant destinés.

Évaluation effectuée par une tierce partie

3.32. Une évaluation effectuée par une tierce partie devrait être effectuée pour les systèmes de sûreté. Son champ d'application et sa portée devraient être définis dans la description du programme d'assurance de la qualité. L'équipe exécutant les tâches d'assurance de la qualité, de vérification et de validation devrait être indépendante de l'équipe de développement. Ces questions sont décrites ultérieurement dans ce Guide de sûreté (par. 3.33 et 4.17).

3.33. L'objectif de l'évaluation par une tierce partie est de fournir un point de vue, concernant l'adéquation du système et de son logiciel, indépendant à la fois de celui du fournisseur et de l'utilisateur (l'exploitant). Une telle évaluation peut être prise en charge par le responsable de la réglementation ou par un organisme agréé par ce dernier. Elle devrait attester du niveau de confiance atteint par le processus de production de l'exploitant et des fournisseurs. La stratégie d'évaluation, la compétence et la connaissance du projet, nécessaires à l'évaluation par une tierce partie pour fournir le niveau de confiance requis, devraient être étudiées soigneusement. De plus, l'évaluation par une tierce partie devrait être convenue entre toutes les parties (responsable de la réglementation, exploitant, fournisseurs) afin que les ressources appropriées puissent être mises à disposition en temps voulu. Certaines des évaluations par une tierce partie devront inclure l'examen du processus (par exemple par

le biais d'audits d'assurance de la qualité et d'inspections techniques). D'autres évaluations par une tierce partie devraient inclure l'examen du produit (par exemple par le biais d'analyses statiques, d'analyses dynamiques, de contrôle du programme et/ou des données et d'analyses de la couverture de tests). L'évaluation du produit final devrait (dans la mesure du possible, par rapport aux contraintes de temps) être effectuée sur la version finale du logiciel. Ceci peut inclure l'évaluation par une tierce partie de produits intermédiaires du processus de développement, tels que les spécifications du logiciel.

DOCUMENTATION

3.34. Pour confirmer le niveau de fiabilité d'un produit logiciel, la pertinence du processus de développement devrait être prouvée. La documentation est essentielle pour l'apport de la 'transparence' et de la 'traçabilité' nécessitées par cette approche. La documentation nécessaire à la conception et à l'implémentation d'un logiciel fiable devrait être claire et précise.

3.35. L'ensemble des documents devrait permettre d'assurer la traçabilité des décisions de conception [14, Q3]. Les documents appropriés devraient être produits à chaque étape du processus de développement. La documentation devrait être mise à jour tout au long du développement itératif, y compris les processus de mise en service et de maintenance courante. Les documents mis à la disposition du responsable de la réglementation devraient être identiques à ceux utilisés par le concepteur. Le concepteur devrait en être informé au tout début du projet.

3.36. La documentation relative aux exigences, à la conception et à la programmation devrait être claire et précise afin que les concepteurs, les programmeurs et les chargés de revue indépendants puissent appréhender complètement chaque stade du développement et vérifier son exhaustivité et son exactitude.

3.37. Une bonne documentation est également essentielle pour la maintenance. Un format adéquat de documentation devrait être utilisé afin de réduire la probabilité d'incohérences et d'erreurs lors de futures modifications liées à la maintenance. Les documents devraient avoir des qualités de facilité de compréhension, précision, traçabilité et exhaustivité, cohérence, facilité de vérification et possibilité de modification, décrites dans les paragraphes 3.38–3.44.

Intelligibilité

3.38. La documentation devrait être compréhensible pour des personnes possédant des niveaux de connaissance et de compétence différents. Le langage utilisé devrait

être clair et, lorsqu'il est formel (par exemple des documents graphiques), devrait posséder une syntaxe et une sémantique bien définies.

Précision

3.39. Les exigences et les descriptions des conceptions devraient être exprimées de manière formelle (avec des sémantiques et syntaxes bien définies), les explications étant fournies en langage naturel. Il ne devrait y avoir qu'une seule interprétation possible de chaque description ou exigence (voir par. 5.1.2 de la réf. [4]).

Traçabilité et exhaustivité

3.40. Le but de l'exigence de la traçabilité est de démontrer que l'implémentation est complète par rapport à la conception et aux exigences du système programmé et de faciliter la détection de fonctionnalités peu sûres dans l'implémentation. La traçabilité de documents de niveau supérieur vers les documents du logiciel contrôle l'exhaustivité, et la possibilité de remonter des documents du logiciel aux documents de niveau supérieur contrôle l'existence d'éléments non spécifiés pouvant être non sûrs. Chaque exigence devrait être identifiée de manière unique.

3.41. Une matrice (ou des matrices) de traçabilité devrai(en)t être mise(s) en œuvre et présenter clairement le lien entre les exigences du système programmé et les éléments qui implémentent chaque exigence du système programmé dans les spécifications de l'ordinateur, la conception du système programmé, les exigences du logiciel, la conception du logiciel et l'implémentation du logiciel. Cette matrice devrait démontrer que toutes les exigences relatives au système programmé ont été complètement testées lors de l'implémentation, l'intégration, l'installation et la mise en service.

Cohérence

3.42. Les documents ne devraient pas contenir de déclarations contradictoires ou incohérentes. Chaque élément d'information devrait posséder un emplacement unique et identifiable dans le document et ne devrait pas être répété ou fractionné sur deux ou plusieurs emplacements. Chaque exigence, élément de conception ou module de programme devrait posséder un identificateur unique (cela aide également pour la traçabilité). La notation, la terminologie, les commentaires et les techniques devraient être utilisés de manière uniforme d'un bout à l'autre de la documentation.

Vérifiabilité

3.43. La vérifiabilité est accrue lorsque les documents sont compréhensibles, sans ambiguïté et permettent la traçabilité. L'utilisation d'un même modèle ou du même langage pour les exigences du système programmé et la conception du logiciel contribuera également à la vérifiabilité, mais ce n'est pas nécessairement la solution à tous les problèmes. Lorsque des langages formels sont utilisés pour spécifier les exigences ou les conceptions, les démonstrateurs de théorème et vérificateurs de modèle peuvent également aider à la vérifiabilité.

Modifiabilité

3.44. Les documents devraient être modifiables, c'est-à-dire que leur structure et leur style devraient être tels que toute modification nécessaire pourra être effectuée facilement, complètement et de manière cohérente et sera facilement identifiable.

4. PLANIFICATION DU PROJET

4.1. Le processus de développement devrait être soigneusement planifié et il devrait être clairement prouvé que le processus a été respecté afin de faciliter l'attribution de l'autorisation pour les systèmes importants pour la sûreté. La planification du projet devrait être documentée dans un plan de sûreté exhaustif et spécifique au système programmé ou sous forme d'un ensemble de plans couvrant tous les aspects du projet. La section 4 décrit chaque aspect dans un plan séparé, mais il est également acceptable de disposer tous les plans dans un même document. Les modifications apportées au système ne devraient pas être exclues et des dispositions devraient être prises afin de pouvoir effectuer des itérations ultérieures d'analyse de sûreté.

4.2. Un plan de développement devrait définir un ensemble d'activités de développement ainsi que les caractéristiques essentielles de chacune des activités. Les autres aspects du projet qui devraient être planifiés sont l'assurance de la qualité, la vérification, la validation, la gestion de la configuration ainsi que la mise en service et l'installation. La figure 1 montre un schéma du processus de développement du système programmé retenu dans ce Guide de sûreté (voir section 2 pour information concernant le choix d'un modèle particulier de développement). La figure 2 illustre la relation des phases de vérification et de validation aux exigences, à la conception et à l'implémentation.

PLAN DE DÉVELOPPEMENT

4.3. Le plan de développement devrait identifier et définir le processus de développement qui sera utilisé pour un projet particulier. Les paragraphes 4.4–4.10 indiquent ce que le plan de développement du système programmé devrait couvrir.

Phases

4.4. Toutes les phases du processsus de développement (fig. 1) devraient être identifiées. Chaque phase, par exemple la conception du système programmé, comporte la spécification, la conception et l'implémentation. L'activité de conception d'une phase fixe les exigences de la phase suivante et la spécification des exigences pour une phase est une partie de l'activité de conception de la phase précédente. L'implémentation, par exemple, est le processus de sélection d'un programme préexistant, incluant les sous-programmes de la bibliothèque, et de création de tout programme supplémentaire nécessaire. La vérification devrait être faite pour chaque phase du développement et avant le début de la phase suivante (voir fig. 2 et par. 4.12–4.18 sur le plan de vérification et de validation).

Méthodes

4.5. Les méthodes à utiliser pour le développement devraient être identifiées. Cette sélection devrait être liée à la description du programme d'assurance de la qualité dans lequel les normes et les procédures sont établies.

Outils

4.6. Les outils à utiliser devraient être précisés dans le plan de développement. Les outils devraient être choisis de manière à faciliter l'application correcte des méthodes, normes et procédures sélectionnées. La planification préalable de l'intégralité du projet aidera à la sélection d'un ensemble intégré d'outils. Les outils devraient être bien qualifiés pour leur fonction relative au développement, à la gestion ou à la vérification du système. L'exactitude des résultats qu'ils fournissent en sortie devrait être garantie par un processus de certification d'outils, au moyen d'une vérification croisée ou d'une méthode de rétro-ingénierie.

4.7. Des outils devraient être utilisés car ils soulagent le personnel des tâches manuelles susceptibles d'entraîner des erreurs, telles la programmation et la vérification. Ils aident à atteindre un niveau de qualité reproductible, garanti et démontré en partie par les preuves apportées lors de leur utilisation antérieure.

Documents

4.8. Les documents à produire lors de chaque phase devraient être identifiés et leur contenu devrait être spécifié. On devrait indiquer tous les emplacements où l'on pourra trouver les exigences, les attributs qualité et les caractéristiques de performance ainsi que les critères d'acceptation qui seront utilisés pour l'intégralité du projet. Les documents devraient fournir la preuve que le projet s'est conformé au plan de développement.

Planning et jalons

4.9. Le planning relatif aux documents devrait être établi et les moments où le projet doit être examiné devraient être précisés. Ce sont les produits d'une tâche de gestion qui englobe :

— l'évaluation de la disponibilité des ressources ;
— l'estimation de la durée de chaque phase, permettant des itérations ;
— l'évaluation des besoins en formation ;
— l'évaluation de l'adéquation des installations et outils disponibles ;
— l'estimation du temps nécessaire pour l'examen et l'approbation par les organismes de réglementation ;
— l'estimation du temps nécessaire pour l'examen du projet aux points clés.

Personnel

4.10. Un plan devrait être préparé pour garantir que le personnel impliqué dans les activités de développement est compétent pour ce qui concerne l'application des normes, procédures et méthodes correspondantes, l'utilisation des méthodes et outils de conception, de programmation et d'analyse ainsi que les pratiques de gestion de la configuration et de contrôle des modifications. Un enregistrement de leur compétence devrait être tenu à jour. Si de nouveaux membres sont ajoutés à l'équipe, ils devraient être étroitement surveillés jusqu'à ce qu'ils aient fait la preuve de leur compétence aux yeux de leurs responsables de ligne.

ASSURANCE DE LA QUALITÉ

4.11. La description du programme d'assurance de la qualité [14, 15] devrait être préparée et réalisée par l'exploitant et fournie pour examen réglementaire (éventuellement pour approbation) avant le début du projet. Un plan d'assurance de la qualité pour le logiciel devrait être produit dès le début du projet. Ce plan devrait couvrir les fournisseurs extérieurs et inclure au minimum ce qui suit :

(1) l'identification des éléments du logiciel et du matériel auxquels s'applique la description du programme d'assurance de la qualité ainsi que celle des normes applicables existantes, des procédures et des outils à utiliser pour le projet ;
(2) pour chaque document à produire, l'indication dans la description du programme d'assurance de la qualité de la personne qui doit l'examiner et l'approuver en vue de sa publication officielle ;
(3) la description de la structure organisationnelle du projet qui doit inclure l'assurance de l'indépendance des vérificateurs responsables de l'assurance de la qualité ;
(4) la description de la compétence et des besoins en formation du personnel impliqué dans le projet ;
(5) le mécanisme d'identification, de communication et de correction de la non-conformité aux normes et aux procédures ;
(6) l'identification de tous les plans nécessaires, comme le plan de développement, le plan de vérification, le plan de gestion de la configuration et le plan de mise en service et d'installation ;
(7) l'indication du nombre et de la portée des audits d'assurance de la qualité en fonction de la classe de sûreté du système ;
(8) les procédures de qualification des outils (voir par. 4.6) ;
(9) le mécanisme de contrôle de la qualité des composants provenant de fournisseurs extérieurs ; si le mécanisme repose sur des procédures de certification externes, comme un essai prototype, la description de ces procédures devrait être également incluse.

PLAN DE VÉRIFICATION ET DE VALIDATION

4.12. La démonstration de l'exactitude et de la sûreté du système requiert diverses activités de vérification et de validation [16]. Dans le contexte du cycle de vie d'un système programmé, la vérification consiste à contrôler la cohérence d'un produit avec le résultat de la phase précédente et la validation consiste à contrôler la cohérence d'un produit avec les objectifs et les exigences de plus haut niveau (Fig. 2).

4.13. La validation devrait être effectuée pour démontrer que le système programmé atteint les exigences fonctionnelles et les exigences globales de sûreté qui lui sont imposées. La validation comporte deux étapes distinctes. La première étape est la validation des exigences relatives au système programmé en fonction des exigences de la centrale et du système (voir par. 2.6). La base de cette validation par rapport aux exigences de niveau supérieur et aux analyses de sûreté devrait être explicitement précisée dans le rapport de validation. La seconde étape est la validation de

l'implémentation du système informatique par rapport aux exigences du système programmé. Les techniques et les procédures de validation explicites devraient être identifiées dans le plan de vérification et de validation.

4.14. La vérification devrait être effectuée pour les produits des phases de développement suivantes (comme défini dans ce Guide de sûreté) :

(1) conception du système programmé ;
(2) exigences du logiciel ;
(3) conception du logiciel ;
(4) implémentation du logiciel.

4.15. Les techniques à utiliser pour vérifier le logiciel devraient être formulées dans le plan de vérification et de validation, les procédures explicites relatives aux techniques devraient être identifiées et leur adéquation par rapport à la classe de sûreté du système devrait être justifiée. Elles sont censées englober une combinaison de techniques, incluant à la fois l'examen statique des documents et l'exécution dynamique de l'implémentation.

4.16. Il faudrait préciser quels enregistrements de résultats de vérification et de validation doivent être conservés pendant toute la durée de vie du système. Les enregistrements devraient prouver que toutes les activités planifiées ont été effectuées, les résultats enregistrés, et les anomalies étudiées et résolues. Les enregistrements devraient être mis à disposition des tierces parties pour audit et revue. Les enregistrements doivent clairement démontrer la traçabilité de tout le travail de vérification et de validation jusqu'aux documents source correspondants.

4.17. L'équipe (ou les équipes) effectuant la vérification et la validation devrai(en)t être identifiée(s) dans le plan. Les tâches de vérification et de validation devraient être attribuées parmi les équipes. Les équipes effectuant la vérification et la validation devraient être indépendantes de l'équipe de développement. La quantité et le type de vérifications et de validations indépendantes devraient être justifiés en fonction de la classe de sûreté du système ; par exemple, l'indépendance financière peut ne pas être exigée pour les systèmes liés à la sûreté. L'indépendance inclut :

(1) l'indépendance technique : le travail devrait être effectué par des personnes différentes en utilisant des techniques et des outils différents ;
(2) l'indépendance hiérarchique : le travail devrait être dirigé et motivé par des personnes différentes. L'équipe de vérification et de validation et l'équipe de développement devraient avoir des lignes de gestion distinctes. Les communications officielles entre les équipes indépendantes devraient être enregistrées ;

(3) l'indépendance financière : il devrait exister un budget distinct comportant des restrictions relatives au transfert des ressources financières entre le développement et la vérification et la validation.

4.18. Le plan devrait inclure un programme d'enregistrement de tous les cas de non-conformité trouvés pendant l'analyse et montrer qu'ils sont convenablement résolus par le processus de contrôle des modifications (par. 4.23–4.24).

PLAN DE GESTION DE LA CONFIGURATION

4.19. La gestion de la configuration est une extension du plan de développement et du programme d'assurance de la qualité et a une importance suffisante pour être décrite séparément (voir aussi Section 5.2 de la réf. [15] et normes existantes relatives à la gestion de la configuration).

Contrôle de version

4.20. Tous les éléments du développement du logiciel, tels que les programmes de compilation, les outils de développement, les fichiers de configuration et les systèmes d'exploitation, devraient être sous contrôle de la gestion de la configuration. Tous les éléments identifiables, tels que les documents, les composants du logiciel ou les structures de données, devraient se voir attribuer une identification unique, incluant un numéro de version. Ces éléments devraient inclure les éléments développés ainsi que les éléments existants qui sont réutilisés ou appliqués de nouveau. Il devrait exister une bibliothèque ou une zone de stockage désignée pour contenir tous les éléments sous contrôle de configuration afin de permettre de retrouver et récupérer tout élément identifié dans n'importe laquelle de ses versions existantes. Il devrait exister une procédure indiquant quand et comment un élément acquis ou développé doit être placé sous contrôle de configuration. Il devrait exister un mécanisme grâce auquel, à des points spécifiques du planning, un ensemble d'éléments sous contrôle de configuration peuvent être identifiés et représentent la base de référence pour le travail suivant. Chaque élément devrait posséder un enregistrement contenant les informations correspondantes, telles que sa date d'achèvement, les modifications incorporées depuis la version précédente, son statut d'approbation actuel et les personnes responsables de sa création, son examen et son approbation. Des procédures d'approbation sont associées à la description du programme d'assurance de la qualité.

4.21. Des liens appropriés sont exigés entre les procédures et les bases de données pour la gestion de la configuration du logiciel et du matériel. Les modifications de la configuration du matériel peuvent affecter les activités de vérification et de validation.

4.22. À tout moment, un vérificateur responsable de l'assurance qualité ou un vérificateur responsable de la réglementation devrait pouvoir demander et recevoir directement tout élément de l'ensemble constituant la description la plus complète et la plus récente du système et du projet (la base de la référence courante). Cet ensemble d'éléments devrait être identique à celui couramment utilisé comme base pour l'analyse ou le développement en cours, à l'exception des travaux en cours qui n'ont pas encore été approuvés ou publiés.

Contrôle des modifications

4.23. Une fois qu'un élément a été placé sous contrôle de configuration, il ne devrait être modifié que conformément à une procédure bien définie incluant une analyse d'impact et cette procédure devrait entraîner une nouvelle version de l'élément plutôt qu'un remplacement de l'élément identifié existant. La procédure de contrôle des modifications devrait conserver les enregistrements des problèmes qui ont été identifiés et ont nécessité les modifications, comment ils ont été analysés, quels éléments ont été affectés, quelles modifications spécifiques ont été apportées pour corriger le problème et quelles versions et base de référence ont été créées pour résoudre les problèmes. La procédure de contrôle des modifications peut également identifier les personnes chargées de l'approbation des modifications si elles se rajoutent à celles ou sont différentes de celles du mécanisme d'approbation pour les éléments de base. En général, une modification devrait mettre en jeu une répétition de tous les processus utilisés pour créer l'élément à l'origine, incluant toutes les analyses, en partant de l'élément affecté de plus haut niveau. Une analyse de régression devrait être utilisée pour identifier les tests requis pour gérer l'enregistrement de vérification et de validation. Les procédures de l'analyse de régression et ses résultats devraient être documentés.

4.24. Après achèvement du développement (et livraison sur le site), un processus de modification différent peut être appliqué étant donné que l'organisation et les personnes responsables de la maintenance peuvent être différentes des développeurs d'origine. Ceci est discuté dans la section 15. Avant l'application du processus de modification, une procédure de modification devrait être écrite et la nécessité de répéter les analyses réalisées durant le développement devrait être considérée. Toute décision de ne pas répéter une des analyses devrait être documentée et justifiée.

PLAN D'INSTALLATION ET DE MISE EN SERVICE

4.25. Après la construction et la validation d'un système en tant que système autonome, celui-ci devrait être intégré aux autres systèmes de la centrale et testé

au sein de l'environnement réel de la centrale. Ce processus d'installation et de mise en service devrait être soigneusement planifié pour coordonner la bonne transition entre le développement et l'utilisation et le transfert des développeurs et vérificateurs vers les utilisateurs et les personnes en charge de la maintenance.

4.26. Le plan d'installation et de mise en service devrait couvrir ce qui suit :

(1) la séquence d'étapes nécessaires à la bonne intégration du système dans la centrale et les états correspondants de la centrale requis pour intégrer de manière sûre le nouveau système ou le système modifié ;
(2) les interactions nécessaires avec l'organisme de réglementation, incluant toute autorisation ou point d'arrêt à respecter avant que le système ne puisse être mis en exploitation ;
(3) les cas de test de mise en service et leur ordre ainsi que les états correspondants de la centrale nécessaires pour confirmer le bon fonctionnement du système dans l'environnement de la centrale ;
(4) les interfaces entre les autres composants et les systèmes nouveaux ou existants de la centrale ainsi que les tests nécessaires pour vérifier le bon fonctionnement de chaque interface ;
(5) la durée de chaque période probatoire ;
(6) la description des enregistrements et des rapports qui seront générés pour décrire les résultats de la mise en service ;
(7) l'initialisation du processus destiné à former et informer les utilisateurs et les responsables de la maintenance ;
(8) le transfert des processus de gestion de la configuration et de contrôle des modifications des développeurs vers les responsables de la maintenance, incluant le mécanisme utilisé pour traiter les anomalies trouvées lors de la mise en service.

5. EXIGENCES RELATIVES AU SYSTÈME PROGRAMMÉ

5.1. Les exigences relatives au système programmé définissent, au minimum, les propriétés fonctionnelles et non fonctionnelles du système programmé nécessaires et suffisantes pour satisfaire aux exigences relatives à la sûreté de la centrale qui ont été définies à un niveau de conception supérieur. La spécification des exigences relatives au système programmé est une représentation du comportement nécessaire du système programmé. La définition précise des interfaces pour l'opérateur, pour le responsable de la maintenance et pour les systèmes externes fait partie intégrante du produit résultant de cette phase. À ce stade, la définition des interfaces se limite aux

propriétés fonctionnelles et non fonctionnelles de ces interfaces ; leur conception ou implémentation peuvent ne pas être encore déterminées.

5.2. L'élaboration des exigences du système programmé est basée sur le résultat des études de conception de la centrale (fig. 1). Les analyses de sûreté, par exemple l'analyse des accidents, les analyses des transitoires ou les analyses de sûreté de la centrale (basées sur des événements initiateurs postulés et des critères de sûreté à satisfaire), constituent une partie essentielle de cette conception. En plus des exigences relatives à la sûreté, certaines exigences supplémentaires non directement associées à la sûreté sont ajoutées à ce stade de la conception, par exemple les exigences de disponibilité. Par conséquent, la définition des exigences relatives au système de sûreté est le résultat d'un effort collectif d'experts issus de différentes disciplines. Cette partie de la conception sort du cadre de ce Guide de sûreté, mais elle fournit les spécifications d'actions de protection et de critères de performances qui doivent être couverts par les exigences relatives au système. Inversement, les exigences relatives au système doivent également être validées par rapport à ces spécifications afin d'être exhaustives et cohérentes.

5.3. L'élaboration de ces exigences est une étape essentielle du processus de développement car des erreurs et des lacunes à ce stade pourraient éventuellement nuire au processus de validation si elles ne sont pas détectées lors du développement. De plus, des défauts dans les exigences relatives au système programmé sont une source potentielle de défaillances de mode commun pour les sous-systèmes redondants contenant un logiciel identique.

RECOMMANDATIONS

Généralités

5.4. Les exigences relatives au système programmé devraient être indépendantes de l'implémentation. Elles devraient considérer le système programmé comme une boîte noire. La prise en compte des résultats de ce travail devrait respecter un formalisme pour les spécifications, compréhensible pour toutes les parties concernées, avec une sémantique et une syntaxe bien définies. Ce formalisme devrait pouvoir être compris par l'exploitant, les fournisseurs et les concepteurs du système programmé et des spécifications du logiciel. Le formalisme peut être pris en charge par des outils aidant à la validation de l'exhaustivité et de la cohérence des exigences.

5.5. Une description claire et précise de ces exigences devrait être rédigée avant le démarrage des étapes suivantes du projet. Cette description devrait être compréhensible

pour tous les experts du responsable de la réglementation et de l'exploitant concernés : ingénieurs procédés, ingénieurs de sûreté, ingénieurs informaticiens et ingénieurs de la centrale.

5.6. La description des exigences relatives au système programmé devrait être simple à utiliser pour la vérification de l'adéquation des spécifications du système programmé et du logiciel et pour la définition des spécifications des essais qui valideront le système lorsque la conception sera terminée.

5.7. La description des exigences devrait être basée sur un modèle précis du système et de son interface avec l'environnement de la centrale. Un exemple d'un tel modèle en termes de paramètres surveillés et contrôlés est donné dans la Section 5 de la réf. [4].

Validation

5.8. Les exigences relatives au système devraient être validées, c'est-à-dire que leur bien-fondé, leur cohérence et leur exhaustivité devraient être établis en liaison avec les spécifications résultant des analyses de la sûreté de la centrale.

5.9. Les outils utilisés pour analyser la cohérence et l'exhaustivité des exigences du système programmé devraient être validés avec un niveau de confiance prouvé être équivalent au niveau requis pour le processus de conception décrit dans les paragraphes 5.10–5.23.

Interfaces système

5.10. Les interfaces système devraient être conçues de manière à faciliter la participation de l'opérateur aux actions de protection telles que l'action manuelle de récupération, les interventions et la réinitialisation manuelle après un arrêt d'urgence. Les données d'entrée et les commandes devraient être reconfirmées par les opérateurs et validées par le système avant d'être utilisées.

5.11. Quand l'interface opérateur utilise un écran, une attention particulière lors de la conception devrait être portée sur les temps de réponse et les moyens de navigation et d'aide [2, 3]. Chaque image ne devrait servir qu'à une seule finalité pour un mode et un contexte donnés. Par exemple, l'utilisation d'une image unique pour l'affichage des valeurs d'un même paramètre dans différentes unités techniques devrait être évitée. Toutes les interfaces devraient être cohérentes et devraient utiliser les mêmes noms et identificateurs. Une analyse de tâche devrait être effectuée pour vérifier l'adéquation entre les dispositions prises pour les actions manuelles de sûreté et les informations affichées associées.

5.12. Les interfaces système devraient également être conçues de façon à faciliter les contrôles en service sans entraîner des actions de protection intempestives.

5.13. Les interfaces système, telles que l'interface avec un réseau de la centrale, devraient être conçues de façon à ne pas interférer avec les fonctions de protection exécutées par des systèmes externes. Les dysfonctionnements et défaillances des systèmes externes ou des systèmes support devraient forcer le système à passer en état sûr.

5.14. Les limites du système, c'est-à-dire l'interface entre le système programmé et la centrale, devraient être définies précisément. En particulier, les interfaces entre le système et les capteurs et les actionneurs, l'opérateur, le responsable de la maintenance et tout autre système externe devraient être spécifiées.

5.15. Le format des données d'affichage, d'entrée et de sortie ainsi que l'organisation des alarmes devraient être soigneusement spécifiés.

5.16. Les données chargées automatiquement, par exemple à partir d'une disquette, devraient subir une vérification par l'ordinateur en les comparant aux données d'origine. Un dispositif de relecture devrait être fourni pour les données d'étalonnage chargées manuellement. La génération automatique de valeurs par défaut par le système sans avertissement devrait être prohibée.

5.17. L'environnement physique et, pour les systèmes de remplacement, les systèmes précédemment installés imposent des contraintes pour les valeurs des variables et des paramètres environnementaux. Ces restrictions devraient être identifiées et documentées. En particulier, la compatibilité avec les systèmes existants de la centrale et la protection contre la propagation des défaillances des systèmes autres que les systèmes de sûreté devrait être étudiée.

Exigences relatives à la sûreté fonctionnelle

5.18. Pour les systèmes de sûreté, les fonctions du système nécessaires et suffisantes pour satisfaire aux exigences relatives à la sûreté de la centrale devraient être spécifiées. Ces exigences relatives à la sûreté fonctionnelle devraient être exprimées en termes de contraintes supplémentaires devant être gérées par le système pour les paramètres de la centrale qui doivent être maintenus dans des limites spécifiées et en termes d'effets que le système doit avoir sur la centrale.

5.19. Une analyse de sûreté devrait également être effectuée pour les systèmes liés à la sûreté afin de déterminer les exigences concernant la sûreté fonctionnelle.

Exigences non fonctionnelles

5.20. Les exigences non fonctionnelles devraient spécifier ce qui suit :

— Les propriétés du comportement du système, c'est-à-dire les contraintes sur l'environnement, la précision, le temps et les performances imposées à ce comportement (les contraintes de temps et de performances devraient rester dans les limites de temps admissibles déterminées par les analyses de sûreté de la centrale).
— Les attributs de sûreté de fonctionnement correspondants, tels que la fiabilité, la disponibilité et la sécurité, exigés pour le comportement du système. En particulier, on devrait envisager la limitation de la fiabilité annoncée pour le système programmé. Les exigences relatives à la sûreté devraient être élaborées à partir de la politique de sûreté définie pour l'environnement du système programmé et devraient prendre en compte les procédures de sécurité à implémenter (voir par. 14.5).
— La robustesse, c'est-à-dire la manière dont le système programmé réagira aux défaillances potentielles à l'interface avec la centrale, comme des pannes de capteurs ou des valeurs d'entrée hors des limites prévues.
— Si des séparations physiques sont nécessaires et, si oui, leur emplacement (par exemple entre les fonctions de contrôle et de sûreté).
— Le type de diversité à introduire dans la conception du système lorsque cela est nécessaire pour satisfaire aux exigences concernant la fiabilité. Les caractéristiques de diversification peuvent être spécifiées en termes de détection, de mesures, de méthodes de déclenchement et de vote diversifiées ; en termes de fonctions différentes pour réagir au même événement initiateur postulé et en termes de composants système diversifiés et indépendants. Les types appropriés de diversité nécessaires devraient être évalués en fonction des exigences de fiabilité pour chacune des fonctions.
— Comment la nécessité éventuelle de remplacement de certaines pièces du système est prise en compte.

Analyse

5.21. Pour les systèmes de sûreté, toutes les exigences fonctionnelles et non fonctionnelles relatives à la sûreté devraient faire ressortir qu'elles sont le résultat d'une analyse de sûreté de la centrale et déterminées par elle.

5.22. En tant que partie prenante dans la démonstration de la sûreté du système, les exigences relatives au système programmé devraient être assujetties à une analyse de défaillance. Les valeurs en sortie du système potentiellement inexactes devraient

être identifiées, leur impact sur la centrale et l'environnement devrait être évalué et l'adéquation avec le principe de défense en profondeur devrait être confirmée.

5.23. Les sources de défaillances de mode commun potentielles devraient être identifiées. Les modes opératoires dans lesquels plusieurs trajectoires similaires des signaux peuvent exister dans plus d'un circuit ou sous-système devraient être identifiés. Dans cette optique, une attention toute particulière devrait être apportée aux parties du système ou aux fonctions qui seront implémentées dans le logiciel.

DOCUMENTS

Contenu

5.24. La documentation produite au cours de cette phase devrait couvrir les éléments suivants :

— les spécifications des exigences fonctionnelles ;
— les spécifications des exigences non fonctionnelles ;
— les spécifications des interfaces avec l'opérateur, le responsable de la maintenance et les systèmes externes ;
— la nécessité de données et d'enregistrements archivés pour les analyses après incident ;
— les spécifications des tests de validation et de leur couverture ;
— le rapport de validation des exigences relatives au système programmé.

Spécifications des exigences fonctionnelles

5.25. Les exigences fonctionnelles devraient être spécifiées sous une forme indépendante des moyens d'implémentation, par exemple en termes de relations fonctionnelles devant être gérées par le système. Ces exigences fonctionnelles devraient spécifier ce qui suit :

— Les relations entre les variables environnementales d'état résultant des contraintes physiques, naturelles ou autres imposées par l'environnement et par les autres systèmes et que le système ne peut pas transgresser.
— Les relations supplémentaires qui doivent être établies et gérées par le système programmé entre les variables surveillées (paramètres de traitement, signaux opérateur) et les variables réglées (sortie vers les actionneurs et les indicateurs) lorsqu'il fonctionne dans cet environnement.

5.26. Ces relations devraient être organisées de telle manière qu'elles reflètent l'organisation fonctionnelle du système, afin d'induire les mises en correspondance nécessaires pour cette organisation de la structure du système programmé et de la structure modulaire du logiciel qui seront définies lors d'étapes ultérieures.

5.27. Les variables devraient être exprimées à l'aide d'une notation technique mathématique standard. Elles devraient être soigneusement définies, par exemple à l'aide de diagrammes, systèmes de coordonnées, signes ou échelles.

5.28. Les différents modes (groupes d'états) et classes de modes dans lesquels le système peut avoir à fonctionner devraient également être clairement identifiés, particulièrement si les interfaces entre le système, la centrale et l'opérateur sont différentes pour ces modes.

5.29. Un formalisme bien défini devrait être utilisé pour décrire les relations fonctionnelles. Des tables de décision contenant les relations fonctionnelles entre les entrées et les sorties peuvent être utilisées dans ce but, en particulier pour la vérification et pour la définition des tests de validation.

Spécifications des exigences non fonctionnelles

5.30. Pour les systèmes de sûreté, ces spécifications devraient découler, de manière probante, de l'analyse de sûreté de la centrale et devraient documenter les contraintes de temps, les contraintes de performances et les exigences relatives à la sûreté de fonctionnement.

5.31. Les exigences relatives à la sûreté de fonctionnement, en plus de spécifier la fiabilité et la disponibilité requises, devraient documenter et justifier ce qui suit :

— les qualifications de composant requises ;
— l'application des exigences de sûreté comme le critère de défaillance unique ;
— l'exigence relative à la séparation physique et fonctionnelle (par exemple, la détection d'un événement postulé par plus d'un paramètre de la centrale et l'indépendance du traitement du paramètre).

Spécifications des interfaces avec l'opérateur, le responsable de la maintenance et les systèmes externes

5.32. Les spécifications fonctionnelles et non fonctionnelles destinées aux interfaces du système devraient être clairement exprimées pour tous les modes distincts possibles de comportement du système. Des modèles spéciaux et des méthodes de

spécification tels que ceux utilisés pour la description des protocoles de communication peuvent être utilisés.

Nécessité de données et d'enregistrements archivés pour les analyses après incident

5.33. La fréquence et la précision nécessaires pour conserver des données et des enregistrements archivés pour l'analyse après incident ainsi que la décision déterminant si de telles données devraient être générées en continu ou seulement en réponse à des conditions spécifiées de la centrale devraient être spécifiées dans un document.

Spécifications des tests de validation et de leur couverture

5.34. Les spécifications des tests devraient couvrir l'intégralité de la fonctionnalité du système et de son interface avec la centrale et devraient faire ressortir qu'elles émanent des exigences fonctionnelles et non fonctionnelles susmentionnées.

5.35. Pour les systèmes de sûreté, les spécifications de tests devraient également spécifier les tests qui sont statistiquement valides au niveau du système et adaptés à la fiabilité requise (voir par. 2.9).

5.36. Les spécifications de tests devraient préciser les résultats attendus.

5.37. Pour les systèmes de sûreté, ces documents de spécifications de tests devraient être vérifiés et approuvés indépendamment par des personnes qui n'ont pas pris part à l'élaboration des spécifications pour les exigences du système. Ces documents de spécifications de tests devraient pouvoir également être revus par les responsables de la réglementation.

Rapport de validation des exigences relatives au système programmé

5.38. Un rapport de validation devrait inclure ce qui suit :

— les spécifications découlant des analyses de sûreté de la centrale en fonction desquelles les exigences relatives au système ont été validées ;
— les étapes exécutées pour effectuer cette validation ;
— les conclusions de cette validation.

5.39. Le rapport de validation devrait, en particulier, pouvoir être revu et mis à la disposition de l'autorité chargée d'accorder les autorisations aussi tôt que possible.

Nature de la documentation

5.40. Des recommandations générales sur la documentation sont fournies dans les paragraphes 3.34–3.44.

6. CONCEPTION DU SYSTÈME PROGRAMMÉ

6.1. La version initiale de la conception du système programmé résulte d'une mise en correspondance (par des moyens systématiques et structurés ou formels) des exigences système devant être satisfaites par le système programmé avec celles devant être satisfaites par une combinaison appropriée de ce qui suit :

— le micrologiciel ou le logiciel prédéveloppé ou préexistant (par exemple le système d'exploitation) ;
— le matériel ou les circuits intégrés spécifiques à l'application ;
— le logiciel à développer ou à élaborer en configurant un logiciel prédéveloppé.

Des améliorations de la conception du système programmé seront réalisées à partir des résultats des activités présentées dans les paragraphes 6.3–6.6.

6.2. La section 7 décrit comment les parties de la spécification de conception du système programmé relatif au logiciel conduisent à la définition des exigences du logiciel. De manière similaire, les aspects de la spécification relative à la conception du système programmé qui se rapportent au matériel devraient être affinés et les normes applicables devront être sélectionnées (par exemple réf. [17]), mais cette activité sort du cadre de ce Guide de sûreté.

RECOMMANDATIONS

6.3. L'architecture sélectionnée du système programmé devrait fournir les interfaces système spécifiées par les exigences relatives au système programmé et devrait implémenter les exigences non fonctionnelles du système programmé, telles que celles se rapportant à la performance et à la fiabilité.

6.4. Pour la conception, la vérification et la validation et pour la maintenance il est nécessaire d'incorporer des exigences 'ajoutées' au document de spécification correspondant au niveau d'abstraction approprié. Celles-ci pourraient inclure ce qui suit :

(1) les exigences résultant des décisions de niveau inférieur concernant la conception du système programmé ou celle du logiciel ;

(2) les exigences résultant de décisions de niveau inférieur concernant la conception de parties du système extérieures au système programmé mais qui peuvent affecter le système programmé (comme la sélection d'instruments de mesure qui nécessitent une compensation logicielle ou un filtrage matériel).

6.5. Si ces exigences se reflètent dans des modifications de la spécification des exigences du système ou de documents de spécifications de niveau inférieur, un moyen devrait être fourni pour gérer le contrôle de la configuration sur ces exigences ajoutées afin qu'une évaluation puisse être effectuée et déterminer si leur couverture par une validation est nécessaire.

6.6. Les moyens servant à démontrer que le système programmé est suffisamment sûr devraient être traités dans le document réservé à la conception du système programmé. Les questions de conception présentées dans les paragraphes 6.7–6.26 devraient être prises en compte.

Séparation entre les aspects sûreté et les aspects non-sûreté

6.7. Le système de sûreté devrait se consacrer à ses fonctions de sûreté. Lorsqu'il est nécessaire et justifié pour les fonctions non destinées à la sûreté de faire partie du système de sûreté, une analyse devrait être effectuée afin de déterminer si l'intégralité du système doit être classée comme système de sûreté et si la fonction de sûreté n'est pas compromise par les autres fonctions.

6.8. Comme la simplicité aide à atteindre la fiabilité, on devrait envisager de regrouper les fonctions et les composants liés à la sûreté et de les isoler des autres systèmes. Ceci peut se faire en enlevant les fonctions et composants non liés à la sûreté du système programmé, à l'aide d'un système programmé réparti ou à l'aide de « pare-feux » appropriés au sein d'un système programmé centralisé.

6.9. Après séparation, les fonctions et composants non liés à la sûreté peuvent être implémentés avec des méthodes moins restrictives ou exploitant plus intensivement les ressources disponibles. De plus, la séparation des fonctions et composants relatifs à la sûreté peut aussi être avantageuse dans la prise en compte du critère de défaillance unique étant donné que les fonctions et composants non liés à la sûreté n'auront pas d'impact sur les fonctions et composants de sûreté. Toutefois, il faudra prendre soin de s'assurer que la séparation est effective.

6.10. La séparation des fonctions et composants de sûreté peut amener un accroissement de la complexité globale du système. Cette complexité supplémentaire, ajoutée

à l'interface pour obtenir la séparation, ne devrait pas annuler le gain attendu de la réduction de la complexité. Plus l'architecture du système programmé est complexe, plus il est difficile d'apporter la preuve de la sûreté.

6.11. Dans l'éventualité d'un accès commun à un composant matériel ou logiciel, comme une fourniture de données, par des fonctions de sûreté et des fonctions non liées à la sûreté, une panne du composant ne devrait pas empêcher l'exécution des fonctions de sûreté. Ce type d'accès commun devrait être évité dans la conception du système programmé s'il peut potentiellement affecter les fonctions de sûreté.

Redondance, découpage en voies et logique de vote

6.12. La redondance devrait habituellement être utilisée pour le système programmé afin de satisfaire aux exigences de fiabilité et de test en fonctionnement. La redondance permet au système de faire face aux défaillances matérielles aléatoires mais ne constitue pas une assurance contre les défaillances de mode commun.

6.13. Pour réduire la probabilité de défaillances dues aux interactions d'un équipement redondant, les sous-systèmes redondants devraient être séparés électriquement et physiquement. Un résultat global correct peut être obtenu en utilisant un mécanisme de vote pour les résultats obtenus par chaque sous-système même lorsqu'un composant est tombé en panne ou lorsqu'un sous-système est mis hors service pour pouvoir réparer le composant défaillant. Comme la logique de vote nécessite elle-même une utilisation commune par tous les sous-systèmes, une mise en mémoire tampon adéquate et suffisante devrait être utilisée pour garantir que la séparation de l'équipement redondant dans les sous-systèmes est conservée. De plus, les sous-systèmes redondants devraient fonctionner de manière asynchrone.

6.14. La nécessité d'une neutralisation lors de la mise en service du système programmé intégré devrait être étudiée car elle peut avoir un impact sur l'interface matériel–logiciel.

Diversité

6.15. Pour réduire la probabilité de pannes de mode commun, la diversité devrait être incorporée à l'architecture du système informatique suffisamment pour satisfaire aux exigences de sûreté et de fiabilité du système programmé global. La diversité des équipements signifie que différent types d'équipements sont utilisés pour exécuter les fonctions redondantes. La diversité fonctionnelle signifie que différentes méthodes sont utilisées pour exécuter une fonction particulière.

6.16. La question du compromis entre l'ampleur de la diversification du logiciel et l'absence de pannes de mode commun n'est pas complètement résolue. L'utilisation de logiciels diversifiés dans des systèmes redondants (décrite en détail dans la section 9) peut réduire la probabilité de défaillance de mode commun par comparaison avec l'utilisation de logiciels identiques. Cependant, il est impossible de prévoir précisément l'impact de la diversité. Des résultats expérimentaux ont montré que pour différentes versions le nombre de défauts identiques varie entre 10 et 100 % du nombre total de défauts du logiciel [18]. Cet aspect, ainsi que les limites de fiabilité pouvant être annoncées pour le logiciel, devrait être pris en compte lors de la détermination de la répartition des fonctions entre le matériel et le logiciel. Un autre aspect à prendre en compte est la complexité de la mise en œuvre d'une logique de vote, spécialement pour le logiciel.

Détection de défaillance et capacité de mise en panne sûre

6.17. Le système programmé devrait être conçu de façon à détecter les défaillances internes ou externes au système qui pourraient l'empêcher de remplir sa fonction de sûreté. Si un telle situation se produit, le système programmé devrait passer par défaut en état sûr, même dans le cas où le système programmé et ses actionneurs en sortie ne sont plus alimentés électriquement. La détermination de l'état sûr n'est pas toujours facile et est susceptible de varier en fonction de la situation dans la centrale (dans certains cas la réaction la plus sûre du système programmé sera de ne pas réagir sauf pour alerter le personnel d'exploitation de la centrale).

6.18. Si un type particulier de défaillance n'est pas suffisamment grave pour nécessiter un passage en état sûr, le système programmé devrait cependant prévenir le personnel d'exploitation et de maintenance de la centrale afin qu'il puisse remédier à la défaillance avant qu'elle ne puisse amener une panne multi-défaillances ayant des implications de sûreté.

Tolérance aux défauts

6.19. L'architecture logicielle et matérielle du système programmé devrait être telle que le système fonctionne selon un mode sécurité prédéfini après apparition d'un nombre limité de défaillances matérielles ou logicielles. Des techniques telles que l'utilisation de sources de données de remplacement et d'architectures redondantes devraient être envisagées. Les limitations de ces techniques et les procédures associées devraient être intégralement décrites dans la conception du système informatique et énoncées dans les manuels de procédures d'exploitation de la centrale.

Surveillance des processus système et surveillance de l'équipement

6.20. L'architecture du système programmé devrait être conçue en prenant en compte la nécessité de surveiller le fonctionnement et les équipements de la centrale, ainsi que l'état de fonctionnement des composants du système informatique lui-même. Par exemple, dans les systèmes de contrôle pour lesquels il est nécessaire de comparer des signaux semblables provenant de plusieurs voies afin de repérer les signaux défaillants, les moyens pour y parvenir sans compromettre la séparation des circuits devraient être déterminés.

Testabilité en fonctionnement

6.21. Un système programmé utilisé dans des systèmes importants pour la sûreté devrait être conçu de manière à pouvoir être testé périodiquement en cours de fonctionnement de la centrale afin de vérifier que le système continue à être exploitable et fiable. Ceci peut nécessiter l'apport de corrections à l'architecture du système programmé.

6.22. La fréquence des tests en service devrait être déterminée en fonction de l'analyse de sûreté dans laquelle l'intervalle de test nécessaire a été déterminé. L'étendue et la couverture des tests en service devraient être déterminées dans la conception du système programmé et devraient être adaptées à leur objectif.

Précision et temps de réponse

6.23. La combinaison du matériel, du logiciel et d'autres éléments constituant le système programmé devrait répondre aux exigences de précision et de temps de réponse demandées afin de satisfaire aux exigences relatives aux performances globales du système. L'analyse de la précision et des temps de réponse devrait prendre en compte toutes les sources possibles de retard ou de distorsion du signal qui peuvent être introduites par tous les éléments de la chaîne depuis les dispositifs de détection de l'état de la centrale jusqu'aux mécanismes d'activation des moyens de contrôle finaux. Ceci inclut les inexactitudes ou les retards introduits par le logiciel et ceux introduits à l'interface matériel–logiciel du système programmé. Il faudrait, en particulier, tenir compte du fait que l'utilisation d'un logiciel ou d'un matériel informatique pour conditionner les signaux de la centrale (comme le filtrage des parasites et la suppression de l'instabilité d'un contact de relais) peut entraîner des temps de réponses plus courts.

6.24. Pour aider à l'identification et à l'analyse des problèmes relatifs au temps de réponse du système, des protocoles de communication déterministes et des techniques

d'allocation des ressources informatiques devraient être utilisés. L'analyse devrait inclure l'étude de demandes faites au système en envisageant le pire des cas (comme des conditions d'accident qui entraînent la modification simultanée d'un grand nombre de signaux et requièrent l'activation de plusieurs commandes). Plus l'architecture des communications du système programmé interne est complexe, plus il est difficile d'apporter une preuve convaincante de la sûreté. Les techniques qui peuvent aider à fournir une telle preuve sont celles qui se rapportent à l'analyse des processus de communication (comme l'analyse des systèmes communicants, les réseaux de Pétri ou les diagrammes de transition d'états [4]).

Exigences non fonctionnelles

6.25. La conception du système programmé devrait également couvrir les aspects non fonctionnels des exigences relatives au système programmé comme la capacité à faire face à des températures extrêmes, à l'exposition aux radiations ou à l'électricité statique, en fonction de l'installation (c'est-à-dire la qualification de l'équipement), l'immunité face aux interférences avec les radiofréquences et aux interférences électromagnétiques comme les parasites et les surtensions électriques introduits via l'alimentation, le signal, les circuits de communication ou les lignes terrestres. Il faut noter que l'immunité face aux radiofréquences et aux interférences électromagnétiques ne peut être démontrée que sur le système installé.

6.26. La nécessité ultérieure du remplacement de certaines pièces du système devrait être envisagée dans la conception du système programmé.

DOCUMENTS

Architecture du système programmé

6.27. La documentation relative à la conception du système programmé devrait contenir une description claire de l'architecture sélectionnée de système programmé. Elle devrait, en plus, inclure une justification de la conception du système programmé vis-à-vis de la sûreté de fonctionnement requise pour le système et de la capacité à remplir toutes les exigences fonctionnelles. Une telle justification peut être fournie, en partie, par la preuve de l'utilisation d'une méthodologie structurée. Également, l'architecture sélectionnée devrait démontrer qu'un compromis entre la simplicité de conception et la capacité à satisfaire aux exigences de performances a été atteint. La justification de la conception du système programmé peut être partiellement obtenue grâce à l'utilisation possible de modélisations et d'analyses, d'itérations de méthodes formelles ou de prototypes. Le résultat de telles analyses devrait être retranscrit dans la documentation sur le système programmé.

6.28. En supplément à l'affinement des exigences fonctionnelles et de la performance du système programmé qui sont définies dès la spécification des exigences du système, la documentation de la conception du système programmé devrait couvrir les exigences issues des paragraphes suivants (par. 6.29–6.40).

Test du système en service

6.29. Les décisions concernant les modalités de tests du système en service (par exemple en fonctionnement ou à l'arrêt, automatisé ou manuel) et concernant la façon dont les besoins en maintenance seront satisfaits (par exemple par étalonnage, isolation et réparation des composants défectueux, utilisation d'appareils de remplacement installés, dispositifs de déclenchement manuel) pourraient nécessiter des interfaces matériel supplémentaires ainsi que des exigences supplémentaires pour la fonctionnalité du logiciel.

6.30. La sélection du support à utiliser pour stocker la logique logicielle (par exemple RAM ou ROM) influera sur l'étendue des vérifications en fonctionnement ou des vérifications périodiques à l'arrêt qui devraient être mises en œuvre pour garantir la permanence de l'intégrité du système programmé.

Évaluation des besoins en dispositifs d'alarmes et en interfaces pour l'opérateur

6.31. Les ergonomes devraient déterminer (par exemple à l'aide d'analyses ergonomiques) la nature des alarmes et de l'équipement à utiliser (par exemple des messages à base de panneaux d'alarme ou à base d'écrans de visualisation). Ils devraient également déterminer le meilleur moyen de fournir aux opérateurs et aux responsables de la maintenance les informations nécessaires pour connaître à tout moment l'état de la centrale et les moyens pour procéder aux ajustements opérationnels lorsque cela s'avère nécessaire. Ces décisions peuvent nécessiter des interfaces matériel et des fonctionnalités logiciel supplémentaires.

6.32. Bien que tous les aspects du système programmé devraient être assujettis à un niveau approprié de contrôle de configuration, on doit envisager le fait que certaines parties du système (par exemple les points de consigne) doivent être plus facilement modifiables que d'autres. Ainsi, il convient que des exigences relatives aux moyens adaptés pour atteindre cela soient spécifiées (voir section 15).

Analyse des interfaces matériel–logiciel

6.33. Une analyse de toutes les interfaces internes (telles que celles des lecteurs et des protocoles de communication) devrait être effectuée et documentée pour démontrer

que toutes les propriétés des interfaces sont spécifiées. Ceci devrait inclure des éléments tels que la définition des informations à transmettre entre les interfaces, l'étude des caractéristiques des protocoles synchrones et asynchrones et la sélection des taux de transfert.

6.34. Les propriétés de sûreté dont on doit faire la démonstration devraient être identifiées (par exemple la 'vivacité' ou la résolution de conflits et l'absence de blocage) et être confirmées en tant que propriétés respectées via l'analyse des propriétés des interfaces.

Traitement du signal entrée–sortie et traitement et stockage des données

6.35. Le traitement du signal peut être effectué au sein du matériel informatique ou du logiciel, ou à l'extérieur du système programmé, afin que les données de la centrale soient disponibles pour le système programmé. Les choix de conception correspondants (avec leur justification) et les caractéristiques des moyens matériels (par exemple mémoire ou disque) et logiques (par exemple bases de données) pour stocker puis accéder aux données devraient apparaître dans la documentation sur la conception du système programmé après une étude soignée de leur adéquation à l'usage envisagé.

Analyse des risques pour le système programmé

6.36. Une analyse des risques devrait être effectuée pour l'architecture du système programmé et la fonctionnalité qu'il contient afin d'identifier tout risque spécifique qui puisse compromettre la fonction de sûreté et indiquer ainsi tout besoin de modification de l'architecture ou d'ajout de fonctions (telles que des autocontrôles) pour minimiser les impacts des risques. Une telle analyse devrait faire partie de la démonstration concernant la sûreté.

Structures de sûreté du système programmé

6.37. En complément des fonctions de réduction des risques ajoutées en réponse à des risques spécifiques identifiés via l'analyse de risques pour le système programmé, des fonctionnalités telles que des horloges de surveillance, la vérification de séquences d'instructions, la réinitialisation de variables et autres mécanismes de détection de défaillances font partie des règles de l'art. Le système devant rester simple, ces fonctionnalités ne sont ajoutées que dans la mesure où elles ne rendent pas le logiciel plus complexe sans raisons valables.

Problèmes de sécurité

6.38. Compte tenu du besoin de maintenir un contrôle strict de la configuration du système programmé, la conception du système devrait déterminer comment une atteinte intentionnelle ou non intentionnelle de la fonctionnalité du système (par exemple accès non autorisé, logiciel non autorisé, virus) peut être empêchée [12, 13]. Ceci doit inclure les détails de contrôles procéduraux ou autres concernant la manière dont les modifications doivent être apportées au système et vérifiées et dont les modifications non autorisées doivent être empêchées. Il devrait y avoir une analyse des menaces visant la sécurité ainsi qu'une justification du niveau de sécurité à implémenter.

Couverture des mécanismes de pannes sûres

6.39. La documentation sur la conception du système programmé devrait spécifier la couverture atteinte par les mécanismes de pannes sûres spécifiés afin de satisfaire aux exigences non fonctionnelles du système programmé. Elle devrait également formuler toute limitation associée aux objectifs de fiabilité.

Exigences relatives à l'intégration du système

6.40. La documentation sur la conception du système programmé devrait contenir les exigences relatives à l'intégration du système pour ce qui suit :

— le micrologiciel ou le logiciel préexistant (par exemple le système d'exploitation) ;
— le matériel ou les circuits intégrés spécifiques à l'application ;
— le logiciel à développer ou à élaborer en configurant un logiciel prédéveloppé.

Ces exigences d'intégration du système sont issues de l'analyse d'interface pour le matériel et le logiciel (par. 6.33 and 6.34) et devraient inclure par exemple la définition des interfaces matériel–logiciel et le traitement des exceptions associées. Ces informations sont nécessaires à l'élaboration de spécifications complètes de tests pour la vérification du système intégré.

Nature de la documentation

6.41. Les documents pour la conception du système programmé devraient faire partie d'un jeu de documents commençant par la spécification des exigences relatives au système. Les parties des exigences relatives au système qui se rapportent au système programmé sont reportées dans les exigences relatives au système programmé. Dans certains cas, les exigences de performances et les prescriptions

fonctionnelles contenues dans les exigences relatives au système sont reportées sans modification, mais le plus souvent elles sont affinées à un niveau d'abstraction inférieur afin d'exprimer les détails ou les décisions prises se rapportant au système programmé.

6.42. Dans les documents relatifs à la conception du système programmé, la présentation des exigences de manière globale et de chaque élément constitutif devrait être non ambiguë et complète et la traçabilité devrait permettre de remonter aux raisons ayant motivé leur inclusion. Elle devrait être cohérente avec les méthodes utilisées pour documenter les exigences fonctionnelles du système, mais à un niveau d'abstraction inférieur.

7. EXIGENCES RELATIVES AU LOGICIEL

7.1. Les exigences relatives au logiciel sont un sous-ensemble des exigences relatives au système programmé qui seront, en finale, implémentées sous forme de programmes informatiques. Toutes les exigences relatives au système programmé devraient être reportées complètement et correctement dans les exigences relatives au logiciel ou dans celles relatives au matériel. Les exigences relatives au logiciel incluront les exigences qui dérivent du choix d'une conception particulière du système programmé. La vérification des exigences relatives au logiciel par rapport aux exigences de niveau supérieur est une étape importante du processus d'autorisation. De ce fait, il devrait être possible pour les développeurs et pour ceux qui n'ont pas établi les exigences (chargés de revue et responsables de la réglementation) de remonter à l'origine des exigences relatives au logiciel et de les vérifier.

7.2. La préparation des exigences relatives au logiciel est un processus intimement lié à la conception du système programmé. Les exigences relatives au logiciel décrivent ce que les composants du logiciel doivent accomplir afin que, lorsque le logiciel est exécuté sur l'ensemble choisi d'ordinateurs avec les périphériques associés, les exigences relatives au système global soient satisfaites. Dans la conception du système programmé, les fonctionnalités peuvent être réparties de différentes manières entre le logiciel et le matériel pour obtenir un bon compromis entre des aspects tels que la performance, le coût, la taille, la simplicité et la compatibilité. Ceci déterminera ce que seront les exigences relatives au logiciel ou au matériel. Les responsables des spécifications devraient être conscients des capacités des ordinateurs, outils et systèmes similaires existants afin qu'il soit possible de satisfaire aux exigences relatives au logiciel pour sa conception et son implémentation.

7.3. Les exigences relatives au logiciel constituent un lien entre le niveau du système et les capacités de l'ordinateur. En tant que tel, elles devraient être compréhensibles en termes de processus dans l'environnement externe et en tant qu'opérations fournies par le programme informatique.

7.4. Si les exigences relatives au système programmé sont suffisamment détaillées et leur documentation suffisamment formelle et si des parties de la conception du système programmé et du programme sont générées par des outils, alors un document séparé regroupant les exigences relatives au logiciel peut ne pas être nécessaire pour ces parties. Cependant, ces parties d'exigences relatives au système programmé à partir desquelles le programme est généré ou réutilisé devraient être considérées comme un ensemble d'exigences relatives au logiciel par rapport auxquelles doit se faire la vérification du programme à venir. Aussi, tout module compilé séparément qui aura été inclus par le générateur de code devrait faire partie de documents distincts pour les exigences relatives au logiciel.

RECOMMANDATIONS

Généralités

7.5. Les exigences relatives au logiciel devraient inclure la description de l'affectation au logiciel des exigences relatives au système, en tenant compte des exigences de sûreté et des conditions de pannes potentielles, des exigences fonctionnelles et opérationnelles dans chaque mode de fonctionnement, des critères de performance, des temps de réponse et des contraintes, de la détection des défaillances, de l'autocontrôle, des exigences de sécurité et de contrôle de sûreté.

7.6. Les exigences relatives au logiciel devraient être correctement documentées et compréhensibles afin de faciliter leur utilisation par les concepteurs du logiciel, les ingénieurs système, les responsables de la réglementation et, si nécessaire, les ingénieurs de procédé et les ingénieurs de sûreté. Elles devraient être écrites sous une forme indépendante de la conception et de l'implémentation qui seront appliquées. Les exigences devraient être divisées et structurées autant que possible afin de faciliter leur compréhension et leur traçabilité par rapport à des documents de niveau supérieur.

7.7. Les exigences relatives au logiciel devraient être analysées afin de déterminer les ambiguïtés, les incohérences et les conditions non définies ; les irrégularités devraient être identifiées, clarifiées et corrigées. Les exigences relatives au logiciel devraient être vérifiables et cohérentes. Les exigences fonctionnelles devraient être

exprimées sous forme mathématique partout où cela est possible. Les exigences non fonctionnelles relatives au logiciel, telles que l'exactitude et la précision minimum, le comportement temporel, l'indépendance des tâches d'exécution et le comportement en cas de pannes sûres, devraient également être exprimées explicitement et, le cas échéant, de manière quantitative.

Fonctions à exécuter

7.8. Le logiciel joue un rôle essentiel dans l'implémentation des exigences fonctionnelles d'un système programmé. Les exigences relatives au logiciel devraient traduire ces exigences fonctionnelles dans des transformations spécifiques d'entrées numériques en sorties numériques. Les transformations requises d'entrées en sorties peuvent se présenter sous la forme d'expressions logiques, de fonctions ou de relations mathématiques, ou d'algorithmes (séquences d'opérations) si les étapes spécifiques de l'algorithme font partie des exigences. Dans le but d'améliorer la traçabilité et l'intelligibilité, la désignation et le codage des variables internes, d'entrée et de sortie devraient, dans la mesure du possible, avoir une signification concrète dans l'environnement du système.

Sûreté

7.9. Les fonctions importantes pour la sûreté du système devraient être présentées en tant que telles dans le document concernant les exigences relatives au logiciel. Pour chaque sortie correspondante, le mode sécurité pouvant être utilisé dans l'éventualité d'une défaillance détectable mais irréparable devrait être indiqué. La conception du système programmé peut également avoir incorporé des relations supplémentaires entre les entrées et les sorties du logiciel ou entre diverses sorties se rapportant à la sûreté. Elles devraient être présentées en tant que prescriptions de sûreté supplémentaires.

Fiabilité et disponibilité

7.10. Les exigences relatives au système programmé incluent des exigences de fiabilité et de disponibilité et ces exigences devraient être reportées dans le logiciel et le matériel. Etant donné la fiabilité requise pour le système programmé et la fiabilité qui peut être envisagée pour le matériel, il peut être nécessaire que certaines exigences pour le logiciel soient incluses de façon que le système puisse remplir ses objectifs de fiabilité (voir Section 6).

7.11. Un objectif global de fiabilité pour le logiciel peut être énoncé, mais il faut bien comprendre que la réalisation d'un tel objectif sera moins démontrable que

l'accomplissement d'autres types d'exigences. Il est extrêmement difficile de démontrer que les exigences quantitatives de fiabilité pour le logiciel ont été respectées. Les méthodes actuellement disponibles ne fournissent pas de résultats pour lesquels la confiance puisse être placée au niveau requis pour les systèmes de la plus haute importance pour la sûreté et, de ce fait, ce Guide de sûreté ne donne aucun conseil concernant l'utilisation de modèles de fiabilité pour le logiciel. Si des postulants proposent l'utilisation de modèles de fiabilité de logiciel pour la certification ou la mise en service, les justifications du modèle devraient être incorporées dans le plan de certification ou de mise en service et recevoir l'agrément des organismes de réglementation.

Interfaces

7.12. Une description des interfaces entre le logiciel et l'opérateur, les instruments (capteurs et actionneurs), le matériel informatique, les autres systèmes ou les autres logiciels pouvant être présents sur le même matériel devrait être fournie. Elle définit la frontière du logiciel.

Contraintes de conception

7.13. Au moment où les exigences du logiciel seront écrites, des choix auront été faits et constitueront une contrainte pour la conception du logiciel et pour son implémentation. Ces choix devraient être documentés ou référencés en tant que partie des exigences relatives au logiciel si nécessaire. Par exemple, le choix d'un matériel informatique constituera une contrainte pour le choix du compilateur et du système d'exploitation et la nécessité d'appliquer le principe de diversité imposera des contraintes pour l'architecture du logiciel. De telles contraintes de conception devraient être justifiées et respecter les conditions de traçabilité.

Modèle

7.14. Les exigences du logiciel peuvent être basées sur une modélisation du système devant être réalisé (Section 5, et Section 5 de la réf. [4]). Dans ce cas, le modèle et son application devraient être bien définis et documentés avec la spécification des exigences. Par exemple, le logiciel de contrôle est parfois décrit à l'aide d'un graphe d'états. L'utilisation d'un graphe d'états devrait être décrite de manière à ce que les exigences des transitions d'états et des fonctions spécifiques à des états particuliers puissent être correctement comprises.

Performances relatives au temps de traitement

7.15. Une description des limites de temps requises pour les transformations effectuées par le logiciel devrait être fournie. Elle peut inclure le temps minimum et maximum entre les entrées et les sorties correspondantes ou entre des sorties liées consécutives ou les caractéristiques de temps de traitement et de taux d'échantillonnage des entrées que le logiciel doit traiter (comme un comportement dépendant de la fréquence d'arrivée de certaines entrées). On peut imposer au logiciel, dans son ensemble, de détecter et récupérer des défauts ou des défaillances puis de récupérer dans un laps de temps spécifié.

Exactitude des calculs

7.16. L'exactitude des calculs pour les informations numériques qui doivent être traitées par le logiciel devrait être spécifiée. Étant donné que le traitement se fera sur un ordinateur possédant une longueur de mot limitée, on doit admettre que des inexactitudes apparaîtront lors des calculs. Le niveau tolérable d'inexactitude devrait être explicitement formulé afin de guider les concepteurs du logiciel dans leur choix de la représentation par le logiciel des données numériques (c'est-à-dire la précision) et de la méthode spécifique de calcul pour les fonctions nécessaires.

Sécurité

7.17. Certains besoins en sécurité du système informatique seront traduits sous forme d'exigences pour le logiciel (comme les vérifications de validité des entrées, des données stockées ou même du programme lui-même). Les besoins en sécurité peuvent également indiquer que certaines informations manipulées par le logiciel, telles que les points de consignes liés à la sûreté, ne seront accessibles qu'aux personnes désignées.

DOCUMENTS

Contenu

7.18. L'objectif principal des documents sur les exigences relatives au logiciel est de constituer la base pour le développement et l'autorisation. De ce fait, ces documents peuvent contenir des aspects liés à la conception du logiciel, à l'autorisation (par exemple des considérations sur les risques, des recommandations pour les fonctions ou les dispositions de sûreté) et à d'autres éléments fournissant la base pour des exigences spécifiques.

Nature de la documentation

7.19. La documentation sur les exigences relatives au logiciel devrait être préparée conformément aux normes dont le formalisme ne doit pas empêcher la lisibilité. La documentation sur les exigences relatives au logiciel devrait être vérifiable et devrait pouvoir être mise à jour. L'utilisation d'un langage de spécification formel peut aider à faire ressortir la cohérence et l'exhaustivité des exigences relatives au logiciel.

7.20. Des conseils détaillés supplémentaires pour la documentation et le style de la documentation sur les exigences relatives au logiciel peuvent être trouvés dans les normes internationales pour les logiciels de haute fiabilité utilisés dans les systèmes de sûreté des centrales nucléaires [5].

7.21. Des recommandations générales sur la documentation sont fournies dans les paragraphes 3.34–3.44.

8. CONCEPTION DU LOGICIEL

8.1. La conception du logiciel est la division du logiciel en un ensemble de modules interactifs et la description détaillée de ces modules. Il est important que la conception soit bien structurée et compréhensible pour les responsables de l'implémentation, de la maintenance, des tests et de la réglementation. La conception devrait inclure, de manière démontrable, toutes les exigences relatives au logiciel et ne devrait contenir aucun élément à risques. La conception s'occupera normalement de l'architecture du logiciel et de la conception détaillée au sein de cette architecture.

8.2. L'architecture du logiciel fait référence à son organisation en modules. Ces modules incluent les processus, les types de données abstraits, les circuits de communication, les structures de données et les modèles d'affichage. Il existe de nombreuses méthodes pour diviser le logiciel en modules avec divers degrés d'importance pour certains types spécifiques de module. Le choix de l'architecture est très important pour la détermination de la simplicité des modules et de leurs interactions mutuelles, qui à son tour détermine la simplicité de leurs spécifications et des tâches de vérification et de validation.

8.3. La manière dont s'imbriquent les modules (l'architecture du logiciel) peut être décrite de plusieurs façons, même au sein d'une seule conception. Ce Guide de sûreté ne recommande aucune méthode particulière de représentation de l'architecture du logiciel.

RECOMMANDATIONS

8.4. Deux niveaux de conception, au moins, sont recommandés : l'architecture du logiciel et sa conception détaillée (pour obtenir des recommandations plus détaillées, voir la section 5.1.1 de la réf. [4] ainsi que les normes et les ouvrages techniques sur le génie logiciel). Les attributs nécessaires de la conception du logiciel sont abordés dans les paragraphes 8.5–8.12.

Limitation de la complexité

8.5. Dans les systèmes important pour la sûreté, une complexité inutile devrait être évitée à tous les niveaux de la conception. Plus la conception est simple, plus il est facile d'atteindre et de démontrer tous les autres attributs. La simplicité permet également d'être plus confiant vis-à-vis de la compréhension complète du logiciel.

Sûreté

8.6. La phase de conception du logiciel devrait s'occuper des risques identifiés lors des analyses précédentes (voir l'analyse dans par. 5.21–5.23 et l'analyse des risques relatifs aux systèmes programmés dans par. 6.36) et des exigences qui ont été identifiées comme étant importantes pour la sûreté. Ces exigences devraient inclure les fonctionnalités nécessaires d'autotest pour détecter les pannes du matériel pouvant apparaître au moment de l'exécution. Le logiciel devrait également surveiller ses propres données et son graphe de contrôle. La propriété de vivacité d'un module matériel–logiciel peut être contrôlée par des mécanismes de détection de défaillances tels que les techniques de « chien de garde ». Lors de la détection d'une défaillance, l'action appropriée devrait être prise en termes de récupération, procédures d'arrêt et messages d'erreur. Toutes les erreurs, permanentes ou transitoires, devraient être enregistrées. La supervision et l'intégrité de l'accès au programme et aux données en particulier devraient être garanties par un logiciel et des techniques de programmation appropriés si elles ne sont pas garanties par le matériel, par exemple en conservant les programmes et les données dans des mémoires programmables effaçables.

Structure compréhensible et modifiable

8.7. Afin de faciliter les revues, l'organisation de l'architecture du logiciel devrait constituer une structure hiérarchique pour fournir des niveaux séparés. Le masquage d'informations (voir la section 3.3.4 de la réf. [4]) devrait être utilisé pour permettre une révision et une vérification par morceaux et faciliter les modifications. L'utilisation de techniques de représentation graphique peut aider à la compréhension. Il est préférable que la conception soit décrite de manière formelle (avec une

sémantique et une syntaxe bien définies), avec des explications données en langage naturel (voir Sections 5.1.3.7, 5.2.3 et 5.2.4, réf. [4], mais il faut noter que d'autres formalismes existent).

8.8. Les interfaces devraient être simples et les modifications prévues devraient être isolées dans un module unique ou dans un petit nombre de modules.

Traçabilité

8.9. Pour faciliter la traçabilité des exigences, chaque élément de la conception, tel qu'un module, une procédure, un sous-programme ou un fichier, devrait posséder un identificateur unique.

Prédictibilité

8.10. L'architecture choisie devrait être déterministe. La conception devrait être sélectionnée de manière à pouvoir prévoir le fonctionnement du logiciel en termes de réponse aux entrées et du temps de réponse correspondant. Une séquence d'opérations fixe et répétée (telle qu'un vote) peut généralement être utilisée plutôt qu'une interruption. Les protocoles de communication devraient être déterministes et ne devraient pas dépendre du bon fonctionnement des systèmes externes.

Cohérence et exhaustivité

8.11. La conception ne devrait contenir ni contradictions ni ambiguïtés. La description des interfaces entre les modules devrait être complète. Les deux parties de chaque interface entre les modules devraient correspondre et il devrait exister, dans la mesure du possible, un ordonnancement partiel et une utilisation de noms de variable, entre les interfaces d'entrée et de sortie des modules, cohérents.

Vérifiabilité et testabilité

8.12. La conception et sa description devraient être telles qu'il soit possible de démontrer que chaque exigence relative au logiciel a été respectée et de vérifier que l'implémentation est correcte par rapport à la conception détaillée.

DOCUMENTS

8.13. La documentation sur la conception du logiciel fournit les informations techniques relatives à l'architecture globale du logiciel et à la conception détaillée de tous

les modules du logiciel. Les contraintes d'implémentation correspondantes devraient également être spécifiées.

8.14. Une démonstration documentée devrait être fournie et prouver que la conception du logiciel s'est occupée des risques identifiés lors des analyses précédentes et des exigences identifiées comme étant importantes pour la sûreté.

Architecture du logiciel

8.15. La décomposition en modules et la mise en correspondance de cette décomposition avec les exigences devraient être décrites et accompagnées des justifications adéquates. L'architecture devrait également prendre en compte les modifications inévitables qui peuvent se produire pendant toute la durée de vie du système, afin que les mises à jour et les mises à niveau du logiciel puissent être effectuées de façon pratique.

8.16. L'existence d'interfaces entre les divers modules du logiciel et entre le logiciel et l'environnement extérieur (documentée dans les exigences relatives au logiciel) devrait être identifiée et spécifiée dans la documentation.

8.17. Si le système comporte plusieurs processeurs et que le logiciel est réparti parmi ces derniers, la description de l'architecture du logiciel devrait alors spécifier quel processus s'exécute sur quel processeur, l'emplacement des fichiers et des écrans de visualisation, etc.

8.18. L'architecture devrait également prendre en compte les contraintes imposées aux modules et interfaces pouvant résulter de la décision d'utiliser le concept de diversité.

Contraintes d'implémentation

8.19. Au stade de la conception, lors du développement du logiciel, il est parfois nécessaire de faire des choix d'ordre technologique. De telles contraintes d'implémentation sont, par exemple, la nécessité de garantir la diversité et les attributs requis pour les langages de programmation, les compilateurs, les bibliothèques de sous-programmes et autres outils support. Ces informations devraient être fournies par la documentation du logiciel. Les contraintes d'implémentation devraient être justifiées ou devraient permettre leur traçabilité jusqu'aux exigences ou contraintes de niveau supérieur.

8.20. Lors de la détection d'une défaillance, la preuve documentée devrait être fournie que les actions prises pour la récupération, les procédures d'arrêt et les messages d'erreur maintiennent le système en état sûr.

Conception détaillée

8.21. Chaque module du logiciel dans l'architecture du logiciel devrait être décrit dans la conception détaillée. Le contenu spécifique de la description dépendra du type de module. En général, la description d'un module devrait définir complètement son interface avec les autres modules et devrait définir complètement la fonction du module et sa fonction dans l'ensemble du logiciel.

Nature de la documentation

8.22. Les recommandations générales concernant la documentation sont fournies dans les paragraphes 3.34–3.44.

8.23. Des diagrammes et des ordinogrammes devraient être utilisés pourvu que la signification des éléments du diagramme soit bien définie. D'autres techniques usuelles de description de la conception sont, entre autres, les diagrammes de trafic des données, les diagrammes de structure ou les méthodes de représentation graphique (voir, par exemple, section 5.1.3.7, réf. [4]).

9. IMPLÉMENTATION DU LOGICIEL

9.1. Les données en entrée de cette phase sont constituées des spécifications de la conception interne et des interfaces des modules du logiciel. Les données en sortie sont les listings contenant le code source et le code exécutable ainsi que les résultats des tests unitaires et des tests des interfaces des modules.

9.2. Le code peut être produit à partir des spécifications du système de plusieurs manières, qui sont essentiellement des combinaisons de deux approches distinctes : le processus de développement classique par le biais des étapes de spécifications et de conception, détaillées dans les sections 5–8, ou, comme mentionné dans la section 5, via l'utilisation d'outils de génération de code recevant en entrée une description, orientée vers l'application, du système avec un langage de programmation évolué. Le choix entre ces deux méthodes dépend des outils et des ressources à la disposition des parties engagées dans le projet et devrait, en particulier, faire le compromis entre la conception et la démonstration de la sûreté de fonctionnement des

outils. Les recommandations dans cette section s'appliquent à toutes les combinaisons possibles des deux approches.

RECOMMANDATIONS

Vérifiabilité et possibilité de revue critique

9.3. L'élaboration d'un code dont on peut démontrer qu'il constitue une implémentation correcte des spécifications du logiciel est une question majeure. Le code devrait pouvoir être vérifié par rapport à ces spécifications. Si la vérification est faite par une personne physique, le code devrait être lisible, comporter les commentaires adéquats et être compréhensible. Des outils validés peuvent être utilisés pour faciliter le processus de vérification du code (voir section 10).

Contrôle des modifications et de la version

9.4. Le code n'est généralement pas élaboré correctement dans sa première version. Les demandes de modifications et les changements apportés à l'implémentation devraient être soigneusement contrôlés et la cohérence entre les versions successives devrait être maintenue.

9.5. L'implémentation peut faire apparaître des omissions ou des incohérences dans les spécifications de la conception du logiciel ou dans les exigences relatives au système programmé. Un système de demande de modification formelle et de contrôle des modifications devrait, de ce fait, être mis en place dans la phase d'implémentation pour traiter ces omissions et ces incohérences. Des enregistrements de ces modifications devraient être conservés et mis à la disposition du responsable de la réglementation. Il devrait également exister un système permettant de maintenir la cohérence entre les diverses versions des modules qui seront élaborés (voir sections 4 et 15) et de garantir une couverture complète des tests des changements apportés.

Langages de programmation

9.6. Pour les systèmes de sûreté, le langage de programmation (ou le sous-ensemble utilisé) devrait posséder une sémantique et une syntaxe rigoureusement définies et documentées.

9.7. Le langage de programmation (ou le sous-ensemble utilisé) devrait posséder une puissance d'expression adéquate. La facilité avec laquelle les spécifications du module peuvent être traduites lors de l'implémentation dépend énormément du pouvoir expressif du langage de programmation.

9.8. Des langages orientés application plutôt qu'orientés machine devraient être utilisés pour le logiciel d'application. Un assembleur ne devrait être utilisé que pour les modules de taille et de fonctionnalité limitées et seulement dans le cas où cela est entièrement justifié par des contraintes de performance en temps réel ou de compatibilité. Sinon, le langage devrait posséder les facilités techniques suffisantes pour prendre en charge la programmation modulaire, pour structurer le code en termes de procédures, sous-programmes ou sous-programmes avec spécifications et appeler des séquences distinctes du programme principal.

9.9. Le langage de programmation et son traducteur ne devraient pas, de par leur conception, empêcher l'utilisation de structures de limitation d'erreurs, la vérification du type du temps de traduction, la vérification du type du temps d'exécution, la vérification des limites de tableaux et la vérification de paramètres. Ces vérifications de temps à l'exécution peuvent, cependant, être rendues inutiles si l'implémentation est mature et entièrement validée ou si elles exigent trop de puissance de la part du processeur au moment de l'exécution.

Utilisation d'outils

9.10. Des outils validés devraient être utilisés car ils soulagent les programmeurs des tâches de réalisation du code qui sont sujettes à des erreurs manuelles, telles que la programmation et la vérification.

9.11. Un ensemble d'outils approprié devrait être sélectionné pour l'implémentation. Ils peuvent inclure des traducteurs et des générateurs de code, des débogueurs, des éditeurs de liens et autres outils de test et de vérification.

9.12. On devrait utiliser des traducteurs minutieusement testés, disponibles et pouvant être mis à jour pendant toute la durée de vie du système. Si un traducteur qui n'est pas complètement testé est utilisé, une vérification et une analyse supplémentaires (voir section 10), effectuées manuellement ou à l'aide d'autres outils, devraient faire la preuve de l'exactitude de la traduction.

9.13. Ces outils devraient être compatibles avec les autres outils utilisés au cours des autres phases du développement.

Diversité

9.14. La ‘diversité logicielle’ est obtenue à l’aide d’une technique de développement dans laquelle deux ou plusieurs versions d’un programme, identiques du point de vue fonctionnel (variantes), sont développées à partir des mêmes spécifications par différents programmeurs ou équipes de programmation dans le but de permettre une détection des erreurs, une fiabilité accrue ou une probabilité plus faible que des erreurs de traducteur ou de programmation puissent influer sur les résultats en sortie. Bien que l’utilisation de la diversité logicielle puisse entraîner une probabilité plus faible de défaillance de mode commun, elle ne devrait pas être considérée comme un remède pour toutes les erreurs. Des erreurs coïncidentes peuvent encore se produire et la probabilité de défaillance de mode commun reste difficile à évaluer. L’utilisation de la ‘diversité fonctionnelle’ devrait être envisagée, car elle peut apporter une réduction notable des erreurs résiduelles. Avec la diversité fonctionnelle, différents moyens sont proposés pour atteindre le même objectif de sûreté. Là où deux ou plusieurs implémentations sont effectuées dans le logiciel, les techniques recommandées pour atteindre un niveau acceptable de diversité logicielle doivent encore être utilisées (par. 6.15 et 6.16).

9.15. La diversité logicielle peut être atteinte à l’aide de différentes techniques, celle d’utilisation de « blocs de récupération » et celle des « N variant » sont les plus usuelles. La diversité logicielle peut être également appliquée aux différents aspects de l’implémentation du logiciel, y compris les tests. On peut utiliser, par exemple, des équipes de programmation indépendantes, des types de solution diversifiés et des environnements de travail différents, des outils et des systèmes support d’exécution (systèmes d’exploitation, compilateurs). Tous ces aspects et techniques devraient être étudiés avant de décider si la diversité logicielle doit être utilisée. Le choix et sa justification devraient être correctement documentés, en tenant particulièrement compte de la complexité supplémentaire introduite par ces techniques.

9.16. La logique de vote peut amener des problèmes de conception complexes dont on devrait tenir compte. Cependant, comme avec les systèmes redondants, la logique de vote devrait s’appliquer aussi loin que possible vers le bas de la chaîne de décision de chaque sous-système distinct afin de maximiser la diversité du sous-système. Par exemple, là où deux ou plusieurs mesures doivent être utilisées pour décider de la nécessité d’un arrêt du réacteur, la logique de vote devrait être appliquée après comparaison de chaque signal avec son seuil de déclenchement, c’est-à-dire que le vote devrait se faire sur le signal à l’état binaire et non pas sur la mesure. Des stratégies de vote telles que le vote en deux sur trois devraient être envisagées pour réduire la probabilité d’action intempestive. Les autres aspects de la technique de vote à étudier sont les tolérances du matériel dans le procédé de mesure (y compris la granularité des

données), la bande passante (asymétrie) autorisées pour la comparaison et la distance entre les points de vote.

9.17. La capacité de détection d'erreur en fonctionnement est un aspect positif de la diversité logicielle dont on devrait tenir compte. Des variantes différentes du logiciel sont plus susceptibles de posséder des sorties différentes dans le cas de vecteurs d'entrée non testés ou de modifications non anticipées de l'environnement, et, de ce fait, un comportement erratique aura plus de chance d'être détecté.

9.18. Tous les types de diversité impliquent un degré d'indépendance. Si on utilise la diversité pour l'équipement, le logiciel ou la diversité fonctionnelle, alors le niveau d'indépendance annoncé devrait être démontré.

Programmation des modules

9.19. Il est essentiel qu'avant de commencer la programmation d'un module les spécifications de sa conception et de ses interfaces soient complètes et disponibles.

Simplicité du code

9.20. Le code de chaque programme d'un module devrait rester simple et facile à comprendre, à la fois dans sa structure générale et dans ses détails. Les structures récursives, les compactages de code, les optimisations et les astuces qui masquent la fonctionnalité du code et sont utilisés pour la convenance du programmeur au détriment de la lisibilité du programme doivent être évités.

Structures de données cohérentes

9.21. Les structures de données et leurs règles d'affectation de noms devraient être les mêmes pour tout le système. Chaque identificateur de structure de données devrait refléter de manière cohérente son type (matrice, variable, constante, etc.), sa portée (locale, partagée, etc.) et sa nature (entrée, sortie, interne, etc.).

Éviter les insécurités de langage

9.22. La plupart des langages de programmation souffrent d'insécurités, ce qui rend difficile, voire impossible, la détection des violations des règles de langage soit par le compilateur soit par l'analyse du texte du programme. Si ces insécurités ne peuvent pas être éliminées en supprimant certaines caractéristiques du langage ou en ajoutant des règles statiques–sémantiques supplémentaires, leur utilisation devrait être évitée, ou au moins faire l'objet de restrictions, être identifiée et minutieusement vérifiée.

Règles de programmation

9.23. Les techniques de programmation recommandées devraient être prescrites dans un ensemble détaillé et approuvé de règles de programmation et les écarts par rapport à ces règles devraient être identifiés lors de la vérification des modules du logiciel. Des conseils relatifs aux règles de programmation peuvent être trouvés dans les normes internationales pour les logiciels à haute fiabilité utilisés dans le systèmes de sûreté des centrales (par exemple [5]).

Autotest et autovérification

9.24. La conception du logiciel inclut les fonctionnalités nécessaires d'autotest pour détecter les pannes du matériel pouvant apparaître au moment de l'exécution. Le logiciel devrait également superviser ses propres données et son graphe de contrôle. Ces fonctionnalités de supervision peuvent ne pas avoir été prévues dans les exigences relatives à la conception du logiciel (voir section 8). Dans ce cas, une demande de modification de ces exigences devrait être faite.

Systèmes d'exploitation

9.25. Seuls des systèmes d'exploitation dont le test complet a donné des résultats satisfaisants devraient être utilisés. Pour les systèmes de sûreté, seuls les systèmes d'exploitation qui se conforment aux recommandations de ce Guide de sûreté devraient être utilisés. L'utilisation du système d'exploitation devrait être limitée aux fonctions indispensables. Ces fonctions devraient être identifiées et devraient posséder des interfaces bien définies. Chaque fonction particulière devrait toujours être appelée de la même manière. Les résultats appropriés et documentés de l'expérience d'exploitation de l'utilisation de ces fonctions devraient être disponibles.

9.26. Des précautions devraient être prises pour garantir que la séparation des fonctions sûreté et des fonctions non liées à la sûreté n'est pas compromise par l'utilisation d'un système d'exploitation commun, d'un autre logiciel support enfonctionnement ou d'un logiciel de communication par réseau.

Testabilité

9.27. On devrait pouvoir accéder à toutes les parties du code exécutable (y compris l'accès pour garantir le bon comportement du programme support à l'exécution et des

mécanismes de supervision des pannes matérielles) pour effectuer des tests, à l'aide d'une ou d'une combinaison des techniques mentionnées dans ce Guide de sûreté, de type 'boîte noire' ou de type 'boîte blanche' (voir, par exemple, la section 9.2.2 de la réf. [4]).

DOCUMENTS

9.28. Le code de chaque programme d'un module devrait apparaître dans une section spécifique d'un document accompagné des informations contextuelles nécessaires pour vérifier l'exactitude de ce programme par rapport à sa spécification.

9.29. Les informations contextuelles fournies devraient être suffisantes pour servir de base pour la compréhension et le test d'un programme d'un module isolé des autres programmes du module ou des autres modules. Ces informations contextuelles sur le module devraient contenir soit une description dupliquée, soit une référence à cette description, la fraction appropriée des spécifications de la conception du logiciel, les assertions logiques ou les conditions préalables et ultérieures auxquelles doit satisfaire chaque programme du module. Elles devraient également identifier les autres programmes et modules qui appellent et sont appelés ainsi que les paramètres et variables d'entrée ou de sortie, accompagnés de leur domaine de validité. Pour faciliter la revue du code, ce document ne devrait pas dépasser une ou deux pages.

9.30. Le choix des langages de programmation utilisés devrait être justifié et documenté. La description de la sémantique et de la syntaxe du langage devrait être complète et disponible.

9.31. Le choix des tous les outils utilisés devrait être justifié et documenté. Le processus ayant servi à la validation des outils devrait également être documenté.

9.32. Des conseils relatifs à la documentation et à la nature des programmes peuvent être trouvés dans les normes internationales pour les logiciels à haute fiabilité utilisés dans le systèmes de sûreté des centrales (par exemple [5]).

10. VÉRIFICATION ET ANALYSE

10.1. La vérification est le processus permettant de s'assurer que les produits de chaque phase de développement (y compris tout logiciel préexistant) satisfont aux exigences de la phase précédente (voir section 2 pour la définition des phases de

développement). L'analyse est le processus consistant à vérifier que les produits de chaque phase de développement ont une cohérence interne et appliquent correctement les techniques choisies. L'objectif de l'équipe de vérification devrait être de fournir un examen complet et approfondi en rapport avec la fiabilité requise.

RECOMMANDATIONS

Généralités

10.2. Diverses techniques de vérification et d'analyse sont disponibles (voir, par exemple, section 8 de la réf. [4] et réf. [16]). Chaque technique fournit une évaluation de la qualité du produit, mais aucune technique ne fournit, à elle seule, l'assurance totale de la qualité. De ce fait, une large panoplie de techniques complémentaires devrait être utilisée. Une inspection et un examen manuels peuvent généralement être effectués sur tous les documents. D'autres formes de vérification et d'analyse sont souhaitables ou même indispensables pour atteindre le degré de certitude désiré. Par exemple, bien qu'une analyse de code statique puisse éliminer la répétition onéreuse d'un programme de tests, elle ne devrait pas être considérée comme pouvant se substituer aux tests. Toutes les formes d'analyse statique sont basées sur des modèles de comportement de programme ; les tests sont essentiels pour explorer les aspects du comportement du programme que ces modèles n'examinent pas.

10.3. Le champ d'application et le niveau de chacune des techniques appliquées pour la vérification et l'analyse devraient être complètement justifiés.

10.4. Les résultats de la vérification fourniront une base de connaissances dans le processus de conception. Des enregistrements du nombre et du type des anomalies devraient être conservés. Ces enregistrements devraient être examinés pour déterminer si des leçons peuvent éventuellement en être tirées et si des améliorations doivent être apportées au processus. Il peut être difficile ou peu judicieux d'implémenter de telles améliorations de processus dans le cadre du développement d'un système spécifique ; les améliorations de processus bénéficieront à des projets ultérieurs de développement de systèmes. Les documents décrivant les processus et les techniques devraient être assujettis à la gestion de configuration qui doit être distincte de la gestion de configuration des documents du système informatique.

10.5. Les techniques manuelles comme les revues, les analyses structurées avec le concepteur, les inspections ou les audits devraient être appliquées à la vérification de toutes les phases du cycle de vie et peuvent être les seules techniques applicables à la vérification des phases qui précèdent la génération du code source. Étant donné que

ce sont des méthodes manuelles, il est important que les moyens d'enregistrement des résultats de telles revues soient étudiés. On peut utiliser des listes de vérification. Cependant, la réalisation des listes de vérification doit viser à optimiser leur utilité (par exemple, la réponse à fournir à un élément des listes de vérification doit être claire et ne doit pas donner lieu à interprétation). Le moyen qui sera utilisé par les vérificateurs pour enregistrer les résultats de leurs examens devrait être indiqué dans le plan de vérification accompagné d'une justification de la méthode choisie.

10.6. L'inspection de la documentation sur la conception du logiciel et sur son implémentation devrait être effectuée avant la définition des tests du logiciel. De cette manière la structure du logiciel peut être dévoilée et la conception de la batterie de tests peut être guidée par le processus d'inspection. Les spécifications relatives aux tests devraient être complètement documentées et revues. La couverture des tests devrait être justifiée. Les examens devraient également vérifier que les contraintes de conception et de codage ont été respectées pour la production du logiciel.

Analyse statique

10.7. Pour établir l'exactitude du code source, les techniques d'analyse suivantes peuvent être utilisées :

— vérification de la conformité par rapport aux contraintes de conception, de codage et de normes ;
— analyse du graphe de contrôle ;
— analyse de l'utilisation des données et du flux de l'information ;
— exécution symbolique ;
— vérification formelle du code.

10.8. Lorsque des exigences relatives au logiciel ont été spécifiées formellement, il est possible d'effectuer une vérification formelle du code. Cependant, la vérification formelle requiert généralement une expertise considérable et, de ce fait, il faut envisager de faire appel à des analystes compétents. Des conseils supplémentaires sur la vérification formelle et le contrôle de la qualité des programmes sont fournis dans la section 5.2.4 de la réf. [4].

10.9. L'analyse statique devrait être effectuée sur la version finale du logiciel.

Stratégie de test et couverture

10.10. Le test est une analyse du logiciel qui met en jeu l'exécution du code objet (de préférence sur l'unité centrale cible ou sur un simulateur). Si cela est appliqué,

le bon fonctionnement de tout le logiciel, des outils de traduction et des éléments du matériel informatique peut être évalué lors du test. Les sollicitations à appliquer au logiciel examiné peuvent être élaborées en concevant un ensemble de tests adéquats. En plus, une stratégie de test permettant un balayage incrémental de tout le logiciel (tel que montant ou descendant) doit être élaborée. Un programme typique de test peut inclure un test initial des programmes du module au niveau le plus bas dans la hiérarchie du logiciel suivi du test des programmes du module en remontant progressivement vers les programmes de niveaux supérieurs. De cette façon, les programmes de niveaux inférieurs déjà testés peuvent être intégrés dans l'environnement de test et peuvent être utilisés pour tester les programmes du module de niveau supérieur.

10.11. Les moyens utilisés pour démontrer qu'une couverture fonctionnelle complète des tests a été atteinte devraient être définis dans le plan de vérification. Il peut être possible de suivre chacun des jeux d'essai en utilisant une matrice de traçabilité comme décrit dans le paragraphe 3.41. Toutes les exigences non fonctionnelles implémentées dans le logiciel, telles que la précision numérique et les performances, devraient être testées.

10.12. Les moyens utilisés pour démontrer que les exigences non fonctionnelles ont été respectées devraient être documentés dans le plan de vérification. Les tests de performance devraient couvrir, par exemple, toutes les exigences de timing, relatives au temps de réponse pour les entrées, au temps mis pour détecter les pannes puis récupérer et à la capacité d'accepter tous les débits d'entrée spécifiés.

10.13. Les objectifs pour le pourcentage de couverture structurelle des tests (tels que couverture des instructions et couverture des branchements à 100%) devraient être indiqués et justifiés dans le plan de vérification. Tout écart par rapport aux objectifs indiqués dans le plan devrait être justifié et documenté.

10.14. Le programme de test devrait porter une attention toute particulière au test des interfaces (telles que module–module, programme de module–programme de module, logiciel–matériel, interne–externe à la frontière du système). Il faudrait s'assurer que tous les mécanismes de transfert de données et que le protocole d'interface fonctionnent de manière satisfaisante.

10.15. Il faudrait étudier les moyens qui serviront au test des conditions d'exception (telles que division par zéro, valeurs hors-limite). Ceci peut nécessiter l'utilisation d'outils de test spécialisés (tels qu'outils d'insertion d'erreurs, émulateurs connectés).

10.16. Les variables d'entrée devraient être testées sur la totalité de leur domaine. Étant donné qu'un test exhaustif est infaisable, on devrait envisager des techniques telles que le partitionnement en classes d'équivalence et l'analyse des valeurs limites

qui peuvent aider à réduire le nombre de tests élémentaires requis pour apporter une couverture de test suffisante sur le logiciel testé.

10.17. Tous les modes de fonctionnement du système devraient être pris en compte lors de la définition des cas de tests.

Préparation et conduite des tests

10.18. Les moyens utilisés pour démontrer qu'une couverture complète des tests a été atteinte devraient être définis dans le plan de vérification (voir section 4). Tout écart par rapport aux objectifs indiquées dans le plan devrait être justifié et documenté.

10.19. Les plans de test devraient être conçus de façon à faciliter le test de régression, en garantissant que les tests sont reproductibles et nécessitent le minimum d'intervention humaine.

10.20. Le personnel qui planifie et conduit les tests devrait recevoir la formation appropriée pour l'utilisation des outils, procédures et techniques de test. Le personnel devrait être indépendant de l'équipe chargée du développement.

10.21. Les équipements de test devraient être étalonnés selon des normes possédant une traçabilité. Les équipements utilisés pour conduire les tests devraient être identifiés de manière unique et consignés dans les procédures de test pour garantir que les tests sont reproductibles et pour aider à la recherche des anomalies.

10.22. Tous les résultats en sortie du système en test devraient être contrôlés. Tout écart par rapport aux résultats prévus devrait faire l'objet d'une enquête. Les résultats de l'investigation devraient être documentés.

10.23. Les enregistrements des tests devraient être conservés. Les enregistrements devraient pouvoir faire l'objet d'un audit de la part d'une tierce partie. La couverture des tests devrait être manifeste, y compris la traçabilité de chaque test jusqu'aux exigences fonctionnelles appropriées.

10.24. Toute anomalie dans les performances d'un test devrait être examinée et, s'il s'avère qu'il faut modifier la procédure de test, une procédure appropriée de contrôle des modifications devrait être appliquée, par exemple en remontant à l'émetteur des procédures de test.

10.25. S'il s'avère que des erreurs existent dans le matériel ou dans le logiciel, toute modification nécessaire devrait alors être apportée sous contrôle des procédures de

modification agréées (voir section 15). La cause des erreurs devrait être analysée, l'absence de détection précoce dans le processus de développement devrait être étudiée et le test de régression approprié devrait être effectué (voir paragraphe 10.19).

10.26. Des conseils supplémentaires sur les tests sont fournis dans les références [4, 5].

Analyse des risques

10.27. Il a déjà été indiqué que la contribution potentielle aux risques, au niveau de la centrale, du système programmé et ses interfaces avec la centrale devait être évaluée à divers stades du développement (les techniques possibles sont évoquées dans la section 8.3.9 de la réf. [4]). Lorsque de tels comportements potentiels critiques sont identifiés, ils devraient être repérés dans la conception du système informatique, la conception du logiciel et le code afin d'identifier les parties de la conception et du logiciel qui nécessitent des dispositions spéciales de conception. De plus, on devrait remonter de ces risques aux exigences et les incorporer, comme il convient, à l'analyse de sûreté de la centrale.

10.28. Il faudrait ensuite vérifier si les critères de sûreté de la centrale ont été respectés dans la phase de conception du logiciel et si la phase d'implémentation du logiciel a affecté la sûreté ou introduit de nouveaux risques. Vu la difficulté de ces vérifications et l'impact qu'elles peuvent avoir sur la conception en termes de modifications ou de fonctionnalités supplémentaires proposées, elles devraient être effectuées pendant que la conception se poursuit et non pas seulement à la fin de l'implémentation.

Évaluation des outils

10.29. Les outils utilisés pour l'élaboration du logiciel appartiennent aux deux grandes catégories suivantes :

(1) Les outils de développement du logiciel, dont les résultats (éventuellement transformés) deviennent une partie intégrante de l'implémentation du programme et qui, de ce fait, peuvent introduire des erreurs. Les générateurs de code, les compilateurs et les éditeurs de lien en sont des exemples.
(2) Les outils de vérification du logiciel, qui ne peuvent pas introduire d'erreurs (mais peuvent être inefficaces). Les outils utilisés pour l'analyse statique et les moniteurs de couverture de tests en sont des exemples.

10.30. Quelle que soit la catégorie à laquelle l'outil à utiliser appartient, il devrait exister une définition précise de sa fonctionnalité. Pour un outil de développement de

logiciel, le domaine d'applicabilité devrait être connu précisément et, pour un outil de vérification de logiciel, l'analyse ou les vérifications qu'il effectue devraient être bien définies.

10.31. Dans tous les cas, un outil devrait posséder une sûreté de fonctionnement suffisante pour garantir qu'il ne compromet pas la sûreté du produit final. De ce fait, un outil de développement de logiciel dont le résultat est utilisé sans vérification ultérieure devrait posséder un niveau de sûreté de fonctionnement extrêmement élevé. Ceci peut ne pas être strictement exigé si ses résultats sont assujettis à une vérification comme décrit dans les paragraphes précédents (10.1–10.28) ou si on utilise une méthode de rétro-ingénierie comme décrit dans le paragraphe 10.32. Pour un outil de vérification de logiciel, les exigences peuvent également être quelque peu atténuées, en raison du fait que ses résultats seront analysés en détail.

Méthode logique inverse

10.32. Dans certaines circonstances, il est possible d'utiliser une méthode de rétro-ingénierie pour confirmer l'exactitude d'une traduction, par exemple d'une description de conception en code source ou d'un code source en code machine. Cette méthode de rétro-ingénierie met en jeu l'application du processus de traduction à l'envers (ou en sens inverse). Sa faisabilité dépend du processus de traduction d'origine ; il faut qu'il soit bien défini, traçable, direct et bijectif. Cette méthode de rétro-ingénierie a été appliquée de manière satisfaisante, et même à l'aide d'un traitement mécanisé, pour le résultat de générateurs de code à partir de la conception. La technique a également été utilisée avec succès pour la reconstruction du code source à partir du code machine où la source était d'un niveau assez faible. La technique ne peut pas être appliquée de manière rigoureuse pour reconstruire des programmes avec des langages de haut niveau, pour lesquels le mappage sur le code exécutable est extrêmement complexe. Toutefois, la technique consistant à traduire les codes source et machine en un langage formel commun puis à utiliser l'aide d'une machine pour comparer les deux versions est performante et, de ce fait, devrait être envisagée.

Évaluation de l'expérience antérieure

10.33. Le résultat de l'expérience antérieure en service du logiciel peut confirmer ses capacités à exécuter les fonctions désirées en plus des autres vérifications. Si de telles informations sont utilisées, une analyse de l'intérêt de son utilisation passée, spécialement sa compatibilité avec le nouvel environnement proposé et le rapport entre cette utilisation et la nouvelle application proposée, devrait être fournie.

DOCUMENTS

10.34. La vérification et l'analyse devraient être documentées. Les documents devraient fournir un ensemble cohérent de preuves confirmant que les produits du développement sont complets, corrects et cohérents.

10.35. Il a déjà été recommandé (section 4) qu'un plan de vérification et de validation soit élaboré au tout début du projet. Ce plan devrait définir l'ensemble de techniques de vérification et d'analyse à utiliser, devrait justifier que cet ensemble fournira les preuves suffisantes et devrait définir les enregistrements, rapports et documents particuliers qui seront préparés. Le plan devrait également couvrir les problèmes de gestion (qui exécutera chaque processus et comment ces processus seront coordonnés entre eux et avec les processus de développement).

10.36. Étant donné que la vérification compare les résultats d'une phase aux résultats de la phase précédente, la préparation de la vérification peut démarrer dès la fin de la phase précédente. Une telle préparation devrait constituer, dans la documentation, la première partie des résultats de la vérification. La documentation devrait comporter ce qui suit :

— les listes de vérification et les réponses désirées ;
— les règles d'épreuve ;
— les cas de test et les résultats attendus (plans de test) ;
— les objectifs de couverture de test et les justifications ;
— les critères d'évaluation des outils.

10.37. Lors d'un processus de vérification, les vérificateurs devraient enregistrer suffisamment d'informations sur le processus de façon à ce qu'il puisse être répété avec la certitude d'atteindre les mêmes résultats. Si des outils de vérification sont utilisés, ils devraient être complètement identifiés (y compris la version et la configuration) et toute entrée ou paramètre de configuration utilisé devrait être enregistré.

10.38. Les résultats d'une vérification devraient être enregistrés pour démontrer que tous les aspects préparés ont été couverts. On devrait indiquer clairement les résultats qui ont atteint les prévisions et ceux qui ne l'ont pas fait (anomalies). Toutes les anomalies décelées par les diverses activités de vérification et les analyses devraient être enregistrées et faire l'objet d'une enquête. Les enregistrements de telles anomalies devraient être conservés, y compris la justification des moyens utilisés pour leur résolution. Lorsqu'il s'avère qu'une modification de la documentation et du code source est nécessaire, les procédures relatives aux modifications décrites dans la section 4 devraient être appliquées.

10.39. Le plan de test devrait être aussi clair et complet qu'il est raisonnablement possible. Certaines des recommandations (telles que la couverture fonctionnelle et structurelle) obligeront à appliquer plus de tests élémentaires que d'autres. Cependant, elles offrent toutes la possibilité d'identifier les tests élémentaires qui devraient être inclus dans le plan de test. Des conseils supplémentaires sur la génération des tests élémentaires sont fournis dans la section 9 de la réf. [4].

10.40. Les plans de test, les procédures, les résultats attendus et les rapports sur les tests devraient être tenus à jour et disponibles pour les audits d'assurance qualité et les évaluations par les tierces parties. Les procédures de test devraient présenter les raisons d'utilisation de chaque test élémentaire et fournir la possibilité de remonter des tests élémentaires aux documents source correspondants. Les résultats de test attendus devraient être indiqués (avec leur méthode de calcul) dans la documentation sur les tests avant l'exécution des tests.

10.41. Les résultats d'analyse de risques devraient être reportés et l'exhaustivité de la couverture devrait être évaluée.

11. INTÉGRATION DU SYSTÈME PROGRAMMÉ

11.1. Cette activité se compose de trois parties :

— le chargement de la totalité du logiciel sur les machines constituant le système programmé ;
— les tests vérifiant que les exigences relatives à l'interface logiciel–matériel sont respectées ;
— les tests vérifiant que la totalité du logiciel peut fonctionner dans l'environnement d'intégration du logiciel–matériel.

11.2. La vérification de l'intégration du système s'effectue après l'intégration et le test du système du concepteur. Dans cette phase, l'objectif est de fournir les preuves démontrant que l'intégration du système a été correctement contrôlée.

RECOMMANDATIONS

11.3. L'activité d'intégration du matériel devrait précéder la phase d'intégration du système programmé (des conseils sont donnés dans la réf. [17]).

11.4. Un plan d'intégration du système devrait être créé à partir des exigences relatives à l'intégration du système qui constituent une partie de la documentation sur la conception du système programmé (voir par. 6.40).

11.5. Une analyse de traçabilité documentée devrait être effectuée et faire partie de l'activité de vérification servant à démontrer l'exhaustivité des exigences relatives à l'intégration du système par rapport à la spécification de la conception du système programmé.

11.6. On devrait s'assurer que seuls les modules matériels et logiciels vérifiés sont soumis à la phase d'intégration du système. Ceci implique la présence d'un contrôle de configuration strict du système intégré, comprenant ce qui suit :

— une liste des versions contrôlée pour tous les modules du matériel ;
— la récupération des bonnes versions de module à partir de la bibliothèque de logiciels.

11.7. Les versions des modules logiciel et matériel utilisées pour les phases d'intégration du système devraient être des versions déjà vérifiées. Les versions non vérifiées peuvent être utilisées à condition qu'il existe une justification documentée adéquate, c'est-à-dire qu'aucune modification ultérieure ne sera requise après vérification ou que l'intégralité de la phase d'intégration sera réitérée si des modifications se révèlent nécessaires.

11.8. La vérification de l'intégration du système devrait être conçue pour vérifier que toutes les interfaces d'intégration (comme matériel–logiciel ou module–module du logiciel) ont été testées.

11.9. Pour les systèmes de sûreté, l'équipe chargée de la conception et de l'exécution des tests de vérification de l'intégration du système devrait être indépendante de celle des concepteurs et développeurs. Les communications liées au projet entre l'équipe de test et les concepteurs devraient être enregistrées. Toute modification apportée aux procédures de test vérifiées devrait être enregistrée et assujettie à une nouvelle approbation.

11.10. Les prescriptions générales de test établies dans la section 10 devraient être respectées.

11.11. Dans la mesure du possible, les tests exécutés dans cette phase devraient être basés sur les principes de la 'boîte blanche'.

11.12. La batterie de tests devrait être justifiée. Lors de l'élaboration des tests élémentaires, on devrait prendre en compte ce qui suit :

— la couverture de toutes les exigences relatives à l'intégration (énoncées et implicites, par exemple les tests de robustesse qui démontrent que le système répond de manière sûre à toutes les conditions d'interface possibles) ;
— toutes les interfaces matériel–logiciel ;
— la couverture de l'intégralité des plages de valeurs (y compris les valeurs hors-limites pour les signaux d'interface) ;
— le traitement des exceptions (par exemple la démonstration d'un comportement acceptable lorsqu'une défaillance du matériel se produit) ;
— le partitionnement en classes d'équivalence et les conditions aux limites ;
— les exigences relatives au timing (telles que le temps de réponse, le balayage des signaux d'entrée, la synchronisation) ;
— la précision.

Des conseils supplémentaires concernant l'élaboration des cas de tests sont donnés dans la section 12.

11.13. L'utilisation de méthodes de vérification, autres que les tests et les outils, pouvant aider à la vérification de l'intégration du système devrait être envisagée.

DOCUMENTS

11.14. La documentation produite au cours de cette phase devrait inclure les éléments suivants :

— un plan de vérification de l'intégration du système ;
— les résultats de l'analyse de traçabilité ;
— la justification des méthodes utilisées pour la vérification de l'intégration du système ;
— la justification du niveau d'intervention dans le fonctionnement normal du système pour étudier son comportement lors du test.

11.15. Le rapport sur la vérification de l'intégration du système devrait couvrir, entre autres, ce qui suit :

— l'examen de la couverture des tests ;
— l'examen des tests et de l'intégration des modules ;
— le récapitulatif des résultats de tests d'intégration matériel–logiciel ;
— l'évaluation des performances de la synchronisation interne ;
— l'examen de la traçabilité.

12. VALIDATION DES SYSTÈMES PROGRAMMÉS

12.1. La validation est le processus consistant à démontrer que le système programmé exécutera la fonction qui lui a été affectée, comme définie par les exigences fonctionnelles et non fonctionnelles (voir section 5). Comme mentionné dans la section 2.6 de la réf. [6], ce processus de démonstration de la sûreté est considéré comme essentiel étant donné que les techniques d'analyse actuelles ne permettent pas de démontrer complètement l'exactitude d'un système. Plus précisément, la validation est considérée comme le test et l'évaluation permettant de déterminer si le logiciel et le matériel du système programmé intégré se conforment aux exigences fonctionnelles et non fonctionnelles. La validation est généralement effectuée hors du site.

12.2. Étant donné la nature numérique des systèmes programmés, il n'est pas possible de tirer profit de l'extrapolation et de l'interpolation pour décider du nombre de tests à effectuer pour démontrer la conformité par rapport aux exigences. Cette limitation oblige le responsable de la validation à justifier la quantité de tests effectués par rapport à la fiabilité requise pour le système. L'utilisation d'un partitionnement en classes d'équivalence et des conditions aux limites peut être envisagée. Il y a également le problème de garantir que certains modes de fonctionnement et certaines interactions entre la centrale et le système programmé (tout ce qui n'est pas testable d'emblée au stade de la validation) seront testés et donneront satisfaction au stade de la mise en service. Enfin, il faut prouver que le niveau de simulation de l'entrée permettant une démonstration acceptable est adéquat.

RECOMMANDATIONS

12.3. Les exigences générales de test établies dans la section 10 devraient être respectées. Les tests devraient être conduits de manière disciplinée et ordonnée, selon la description du programme d'assurance de la qualité et les procédures qui sont contrôlées au sein du régime d'assurance qualité conformément à la réf. [5]. Des conseils supplémentaires sont donnés dans la section 2.6 de la réf. [6] et dans la réf. [7].

12.4. L'équipe qui conçoit les tests de validation devrait être indépendante des concepteurs et des développeurs. Les communications liées au projet entre l'équipe de test et les concepteurs devraient être enregistrées. Toute modification apportée aux procédures de test vérifiées devrait être enregistrée et assujettie à une nouvelle approbation.

12.5. Les tests exécutés à ce stade sont essentiellement des tests de type 'boîte noire' basés sur les exigences relatives au système programmé. Toutefois, une analyse des

fonctionnalités ajoutées au cours des différentes étapes de conception devrait être effectuée afin d'identifier les fonctionnalités qu'il est souhaitable de tester au niveau système (par exemple par injection de signaux de test à la frontière du système), par exemple la validation des fonctions de pannes sûres qui doivent réagir correctement à la suite d'une détection de signaux d'entrée de mauvaise qualité ou hors limites.

12.6. La batterie de tests devrait être justifiée. Lors de l'élaboration des tests élémentaires, on devrait prendre en compte ce qui suit :

— la couverture de la totalité des exigences (y compris les tests de robustesse et les dispositions de sécurité) ;
— la couverture de l'intégralité des plages de valeurs (y compris les valeurs hors limites pour les signaux d'entrée) ;
— le traitement des exceptions (par exemple la démonstration d'un comportement acceptable lorsqu'une défaillance en entrée se produit) ;
— le partitionnement en classes d'équivalence et les conditions aux limites ;
— les exigences relatives au timing (telles que le temps de réponse, le balayage des signaux d'entrée, la synchronisation) ;
— la précision ;
— toutes les interfaces (telles que l'interface matériel-logiciel dans l'intégration du système et les interfaces externes lors de la validation) ;
— les tests de stress et de surcharge (par exemple pour déceler les effets de falaise) ;
— tous les modes de fonctionnement du système programmé, y compris la transition entre les modes et la récupération après une panne d'alimentation.

12.7. Une analyse de traçabilité devrait être effectuée pour démontrer que les exigences de validation (pour les tests ou l'évaluation) sont exhaustives par rapport aux exigences du système programmé.

12.8. Le matériel du système soumis aux tests de validation devrait être complètement représentatif de la configuration finale du système programmé sur l'installation du site et le logiciel du système devrait être identique.

12.9. Le système devrait être soumis à des tests portant sur un domaine étendu d'entrées statiques et d'entrées dynamiques. Il est important de tester tous les éléments du système à l'aide de scénarios représentant la réalité et mettant en jeu toutes les entrées possibles (test dynamique). Cependant, comme il n'est ni raisonnablement faisable ni sûr de tester le comportement d'un système de sûreté en utilisant des scénarios réels d'accident sur la centrale, le système devrait être testé en utilisant une simulation de scénarios d'accident. Un logiciel de test peut être utilisé pour enregistrer les résultats.

12.10. Les tests dynamiques devraient être basés sur une analyse des transitoires de la centrale induits par des évènements initiateurs postulés. Les profils de test devraient être représentatifs des variations prévues des paramètres de la centrale qui solliciteront le système programmé. Le nombre de tests exécutés devrait être suffisant pour assurer la sûreté de fonctionnement du système.

12.11. La possibilité de soumettre le système programmé à de tests statistiques devrait être étudiée. Le but est de fournir une estimation de la fiabilité du système programmé en le soumettant à une série de tests statistiquement valables (voir section 9.3.7, réf. [4]). Ces tests devraient être sélectionnées aléatoirement à partir du domaine d'entrée opérationnel du système programmé.

12.12. Des tests purement aléatoires, impliquant qu'ils soient choisis parmi toutes les combinaisons possibles d'entrées, peuvent également être utilisés car ils fournissent un moyen pratique de générer un grand nombre de tests élémentaires pouvant révéler des effets secondaires inattendus. Lorsque la combinaison d'entrées n'est pas incluse dans le domaine d'entrée opérationnel du système programmé, le test correspondant est défini comme étant un test de robustesse.

12.13. Pour les tests dynamiques et statistiques, le nombre de tests appliqués devrait être proportionné aux exigences de fiabilité relatives au système programmé. On devrait fournir la preuve que la méthode de détection d'erreur est appropriée.

12.14. Des tests devraient être effectués pour confirmer l'hystérésis et l'exactitude des points de consigne.

12.15. Les manuels d'exploitation et de maintenance devraient être validés, dans la mesure du possible, au cours de cette phase.

DOCUMENTS

12.16. La documentation de la phase de validation doit inclure les éléments suivants :

— les mises à jour possibles du plan de validation ;
— l'analyse des fonctionnalités ajoutées au cours des divers stades de la conception ;
— l'analyse de traçabilité démontrant que les exigences de validation (pour les tests ou l'évaluation) sont exhaustives par rapport aux exigences du système programmé ;

— les résultats de la validation ;
— le plan des tests statistiques, le planning et les résultats ;
— l'analyse des transitoires de la centrale induits par des événements initiateurs postulés à utiliser dans les tests statistiques ;
— le plan des tests aléatoires, leur planning et leurs résultats.

13. INSTALLATION ET MISE EN SERVICE

13.1. Après avoir été livré sur le site, le système programmé doit être installé dans la centrale. C'est un processus progressif mettant en jeu la bonne installation des divers éléments de l'équipement à la place qui leur est affectée, la connexion à l'alimentation électrique et la répétition de certains des tests effectués dans les locaux du fabricant afin de s'assurer que le système programmé n'a subi aucun dommage pendant son transport et son installation. Cette phase est suivie de la connexion des câbles de la centrale au système programmé : ce peut être des câbles de transmission et des câbles de contrôle-commande. Après leur connexion, il faudrait démontrer que chacun des câbles a été connecté aux broches appropriées.

13.2. La mise en service est le processus au cours duquel on fait fonctionner les systèmes et composants déjà construits de la centrale nucléaire et on vérifie qu'ils sont conformes à ce qui était prévu par la conception et qu'ils satisfont aux critères de performance. La mise en service comporte des tests non nucléaires et des tests nucléaires. Au cours de cette phase, le système programmé est progressivement intégré aux autres composants et aux autres éléments de la centrale (intégration du système). Le test au sein de l'environnement de la centrale est une partie importante de la mise en service des systèmes programmés. En particulier, les modes de fonctionnement et l'interaction entre le système programmé et la centrale qui ne peuvent pas être testés d'emblée au stade de la validation devraient être testés lors de la phase de mise en service.

13.3. Les activités d'installation et de mise en service peuvent s'effectuer soit sur une nouvelle centrale soit lors d'une modification de la centrale (par exemple lors d'une adaptation ou d'une mise à niveau). Les recommandations données dans les paragraphes 13.5–13.10 s'appliquent dans les deux cas, nouvelles centrales ou modifications de centrales existantes.

13.4. La validation du système programmé par rapport aux exigences relatives à la sûreté de la centrale est la dernière étape du processus de vérification et de validation.

RECOMMANDATIONS

13.5. Le système complet ne devrait pas être installé dans la centrale tant que la phase d'intégration du système programmé et de validation n'est pas terminée. Cependant, si cela n'est pas possible, la justification du bien-fondé des tests effectués après installation devrait être fournie. La démonstration de la sûreté du système installé devrait être effectuée au même niveau que celui qui aurait prévalu dans les locaux du constructeur (voir sections 8 et 9 de la réf. [4] et sections 7 et 8 de la réf. [5]).

13.6. Il faudrait envisager de répéter certains des tests ou tous les tests évoqués dans les sections 10–12, étant donné que les équipements, et par conséquence le logiciel, ont pu être endommagés lors de l'installation. Il faudrait prêter une attention toute particulière au test des interfaces système externes et à la confirmation d'un bon comportement avec les équipements aux interfaces. Les tests effectués lors de la phase de mise en service devraient confirmer, partout où cela est faisable, les tests précédents vérifiant l'exactitude des réponses normales et exceptionnelles des interfaces.

13.7. Le système programmé devrait être assujetti à une période probatoire supplémentaire sur site pendant laquelle le fonctionnement, le test et la maintenance du système devraient être aussi représentatifs que possible des conditions réelles d'exploitation. Le cas échéant, le nouveau système doit fonctionner en parallèle avec l'ancien système pendant la période probatoire, c'est-à-dire tant que la confiance acquise vis-à-vis de l'adéquation du nouveau système n'est pas suffisante. La validation des manuels d'exploitation et de maintenance devrait être terminée au cours de cette phase. Au cours de cette période, le système devrait être soumis aux activités courantes d'essais et de maintenance. Les défauts décelés et les actions correctives entreprises devraient être consignés et conservés dans un journal.

13.8. Il faut admettre qu'au fur et à mesure que le système programmé est progressivement intégré dans la centrale, il arrivera un moment où le système programmé sera contraint de remplir sa fonction de sûreté. Tous les tests praticables du système programmé nécessaires pour permettre une mise en service et assurer un fonctionnement continu doivent être terminés avant que ce moment arrive. Aussi, tout le matériel ou logiciel nécessaire à la maintenance du système devrait être en place avant que le système ne doive assumer sa fonction de sûreté.

13.9. L'équipe réalisant les tests de mise en service devrait être indépendante des constructeurs du système de sûreté programmé. On devrait également s'assurer que les personnes impliquées dans le programme de mise en service sont compétentes pour les tâches qui leur sont attribuées.

13.10. Un contrôle strict de la configuration du système informatique (matériel et logiciel) devrait être maintenu au cours du programme de mise en service. Toute modification requise dans cette phase devrait être soumise à un processus de modification documenté de manière formelle (voir la rubrique des modifications après livraison dans la section 15).

DOCUMENTS

13.11. Une documentation suffisante devrait être produite pour démontrer l'adéquation du programme de mise en service à satisfaire complètement le système de sûreté programmé installé. Une démonstration documentée de l'adéquation de la couverture de tests, incluant une traçabilité des tests permettant de remonter aux exigences initiales (exigences relatives au système de sûreté), devrait être fournie par l'équipe chargée de la mise en service. La documentation justifiant l'adéquation du programme de mise en service devrait être conservée par l'opérateur de la centrale et devrait être disponible pour une évaluation par une tierce partie. Les documents produits au cours de cette phase devraient inclure les justifications relatives aux éléments suivants :

— l'adéquation de chaque test d'intégration et de validation du système effectué après l'installation ;
— le nombre de répétitions des tests avant installation (parfois dénommés essais en usine ou essais d'acceptation) à effectuer au cours de cette phase ;
— la couverture des tests de mise en service, incluant la traçabilité des tests permettant de remonter aux exigences initiales (exigences relatives au système informatique), tous les enregistrements de mise en service et un rapport d'achèvement plus un enregistrement des anomalies et des échecs en cours de test, incluant une explication de leur résolution ;
— la capacité du système programmé à remplir la fonction de sûreté appropriée dans toutes les phases de la mise en service ; en plus, la justification devrait démontrer que tous les tests praticables du système programmé nécessaires pour permettre une mise en service et assurer un fonctionnement continu ont été exécutés avant que le système ne doive remplir les fonctions de sûreté appropriées.

14. EXPLOITATION

14.1. La phase d'exploitation d'un système programmé suit l'installation, la mise en service et toute approbation réglementaire d'utilisation. Le système fait alors partie

de la centrale et est exploité par l'exploitant de l'installation. La phase d'exploitation d'un système particulier se poursuit jusqu'à ce qu'il soit retiré ou remplacé (éventuellement par une version modifiée comme décrit dans la section 15).

14.2. L'exploitation d'un système programmé comportera des activités de maintenance afin de garder l'équipement en bon état et de réparer les composants défectueux. Les processus qui prennent en charge l'exploitation des systèmes de sûreté des installations nucléaires peuvent potentiellement compromettre l'aptitude de ces systèmes de sûreté à remplir leur fonction s'ils ne sont pas d'une intégrité suffisamment élevée.

RECOMMANDATIONS

Période probatoire

14.3. Comme recommandé dans le paragraphe 13.7 pour les nouveaux systèmes, à la suite d'une modification, une période probatoire devrait être imposée au cours de laquelle les tests en service du système programmé doivent se dérouler avec une fréquence accrue.

Actions correctives

14.4. Après défaillance d'un composant matériel, les actions correctives devraient se limiter au remplacement un pour un du matériel et au rechargement des modules de logiciel existants ; ces actions ne devraient comporter aucune modification (voir la rubrique sur les modifications après livraison de la section 15).

Sécurité

14.5. Sur la base de la politique de sécurité définie pour l'environnement du système programmé, les procédures de sécurité appropriées — par exemple la gestion des mots de passe — devraient être mises en œuvre (par exemple se prémunir contre les accès non autorisés et les virus). Voir par. 5.20 pour les exigences relatives à la sécurité et par. 6.38 en ce qui concerne les mesures de sécurité dans la conception du système programmé.

14.6. Des dispositifs de stockage sécurisé et des contrôles procéduraux devraient garantir que seules les versions de logiciel autorisées sont chargées sur l'équipement de la centrale. Le bon comportement du système programmé devrait être démontré avant qu'il ne soit remis en service.

Données d'étalonnage

14.7. Les données d'étalonnage devraient être d'une précision suffisamment élevée afin de ne pas dégrader la fiabilité du système programmé. Pour les systèmes de sûreté, de telles données devraient être générées automatiquement par un système développé selon des normes identiques à celles du système programmé. Dans les cas où le système de génération de données n'a pas été développé selon ces normes, les données d'étalonnage devraient être vérifiées à l'aide de diverses méthodes appliquées par un groupe indépendant.

14.8. On devrait exiger que l'opérateur valide les données entrées conformément à une procédure agréée avant de passer à l'élément suivant. Toutes les données entrées devraient être archivées séparément et vérifiées par une partie indépendante avant que le système de sûreté ne soit remis en exploitation.

14.9. Un test approprié devrait être effectué sur le sous-système concerné à la suite de la modification des données d'étalonnage afin de démontrer son bon fonctionnement.

DOCUMENTS

14.10. Des enregistrements complets et précis de l'exploitation du système devraient être conservés par l'exploitant et devraient être disponibles pour une évaluation par une tierce partie. Ces enregistrements devraient inclure les informations relatives à toutes les activités de maintenance (y compris les actions préventives et correctives), aux tests en service et aux anomalies observées lors de l'exploitation du système.

14.11. Les incidents et anomalies associés au système programmé (y compris les difficultés rencontrées pour l'exploitation et l'utilisation des manuels d'entretien) devraient être enregistrés selon les procédures qui doivent être mises en place dans ce but. Les résultats des enquêtes sur les incidents et des actions correctives entreprises devraient être indiqués. Les actions correctives impliquant des modifications du logiciel sont assujetties à un processus de contrôle strict des modifications comme décrit dans la section 15.

14.12. Pour les systèmes de sûreté, si le système de génération des données d'étalonnage n'a pas été développé selon les mêmes normes que le système programmé (voir par. 14.7), le processus de génération des données d'étalonnage devrait être décrit et faire ressortir qu'il est différent. Les groupes effectuant les calculs requis

devraient être identifiés et des procédures garantissant une diversité maximale devraient être en place. Tous les calculs devraient être documentés et conservés pour un audit ultérieur.

15. MODIFICATIONS APRÈS LIVRAISON

15.1. Au cours de la phase opérationnelle, il est nécessaire de s'assurer que la sûreté fonctionnelle du système programmé est maintenue pendant et après les modifications. Comme mentionné au paragraphe 4.24, les procédures de contrôle des modifications qui se rapportent spécifiquement aux problèmes de sûreté de la centrale devraient par conséquent être mises en place. Il est également important que l'impact sur la sûreté des modifications soit analysé par un personnel issu des disciplines appropriées. De ce point de vue, les modifications apportées aux systèmes programmés ne diffèrent pas des autres modifications de la centrale. Toutefois, certains problèmes spécifiques s'appliquent uniquement aux systèmes programmés et à leur logiciel et sont traités dans cette section.

RECOMMANDATIONS

15.2. Les concepteurs et les opérateurs de la centrale devraient s'assurer que les procédures adéquates de contrôle des modifications sont en place, y compris les structures organisationnelles et les procédures appropriées pour l'examen et l'approbation des aspects sûreté de la modification. Ces procédures devraient être en place au début du projet.

15.3. Toutes les modifications devraient être examinées et classées en fonction de leur importance vis-à-vis de la sûreté. Celles qui ont le plus d'impact sur la sûreté devraient éventuellement être soumises aux organismes de réglementation, en particulier celles qui seraient susceptibles de modifier les conditions et limites opérationnelles précédemment approuvées.

15.4. Il devrait exister une procédure selon laquelle des experts indépendants des concepteurs et des développeurs de la modification doivent évaluer le bien-fondé de la modification proposée et de son implémentation. Les modifications qui devraient être couvertes par les procédures de modification comprennent les modifications de logiciel, les modifications de matériel et les modifications d'outils.

15.5. Les procédures de modification devraient garantir qu'une justification est fournie pour chaque modification (voir par. 15.3).

15.6. Les modifications apportées au système programmé lors du fonctionnement en ligne, et en particulier à son logiciel, ne devraient être autorisées que si elles sont étayées par une justification détaillée, et aucun équipement ne doit être fourni pour cette opération. La modification de paramètres pouvant demander des évolutions au cours de l'exploitation de la centrale (tels que des seuils d'arrêt d'urgence et des constantes d'étalonnage) devrait être effectuée à l'aide de moyens ayant démontré qu'ils sont appropriés à cet usage. Le degré d'évolution apporté par ces moyens devrait être limité au domaine justifié dans l'analyse de sûreté de la centrale.

15.7. Un contrôle strict de la configuration devrait être maintenu tout au long du processus de modification, en particulier pour résoudre tout conflit résultant de modifications entreprises simultanément. Seuls les éléments ayant suivi le processus complet de modification devraient être installés sur les équipements de la centrale. Les procédures de modification devraient prévoir des dispositions pour les cas où le dépannage ou les tests nécessiteront des modifications temporaires pendant que le système programmé n'est pas en ligne.

15.8. L'installation du logiciel modifié dans un sous-système redondant de différents sous-systèmes devrait être planifiée et effectuée de telle manière que les effets de la dégradation temporaire du système soient minimisés. L'installation du logiciel modifié dans le système de sûreté devrait être faite progressivement par sous-système afin de réduire les effets de cause commune (c'est-à-dire un sous-système à la fois).

DOCUMENTS

15.9. Pour chacune des modifications proposées, les informations suivantes devraient être fournies, suivant le cas :

— la raison de la modification ;
— la description fonctionnelle de la modification ;
— l'évaluation, du point de vue sûreté, de la modification démontrant que la sûreté de la centrale n'est pas compromise par les modifications ;
— la description détaillée de la modification de la conception, avec une analyse d'impact qui couvre la totalité de la portée de la modification proposée, y compris les éléments de la centrale qui peuvent être affectés ;
— les rapports sur la vérification et la validation ainsi que sur l'évaluation par une tierce partie, incluant la justification de leur champ d'application ainsi que celle de l'analyse de régression ;
— la justification de la méthode d'installation proposée (voir par. 15.8) ;
— le rapport sur les tests effectués sur le site.

15.10. Tous les documents relatifs aux modifications devraient être datés, numérotés et classés dans la liste des contrôles de modification du logiciel.

15.11. Un enregistrement chronologique devrait être établi et conservé. Il devrait fournir le détail de toutes les modifications et de tous les changements, y compris les références à la demande de modification ou de changement, l'analyse d'impact, la vérification et la validation des données et des résultats ainsi que tous les documents concernés par les activités de modification et de changement.

15.12. La documentation relative aux modifications devrait être mise à la disposition du responsable de la réglementation qui peut vouloir approuver le processus des modifications et leur implémentation.

RÉFÉRENCES

[1] AGENCE INTERNATIONALE DE L'ÉNERGIE ATOMIQUE, Sûreté des centrales nucléaires : Conception, collection Normes de sûreté n° NS-R-1, AIEA, Vienne (en préparation).

[2] AGENCE INTERNATIONALE DE L'ÉNERGIE ATOMIQUE, Système de protection et dispositifs associés dans les centrales nucléaires, collection Sécurité n° 50-SG-D3, AIEA, Vienne (1981).

[3] AGENCE INTERNATIONALE DE L'ÉNERGIE ATOMIQUE, Systèmes d'instrumentation et de commande liés à la sûreté dans les centrales nucléaires, collection Sécurité n° 50-SG-D8, AIEA, Vienne (1985).

[4] INTERNATIONAL ATOMIC ENERGY AGENCY, Software Important to Safety in Nuclear Power Plants, Technical Reports Series No. 367, IAEA, Vienna (1994).

[5] COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE, Logiciel pour les calculateurs utilisés dans les systèmes de sûreté des centrales nucléaires, Norme n° 880, CEI, Genève (1986).

[6] EUROPEAN COMMISSION, European Nuclear Regulators' Current Requirements and Practices for the Licensing of Safety Critical Software for Nuclear Regulators, Rep. EUR 18158 EN, Office for Official Publications of the European Communities, Luxembourg (1998).

[7] ATOMIC ENERGY CONTROL BOARD, CANADA; DIRECTION DE LA SÛRETÉ DES INSTALLATIONS NUCLÉAIRES, INSTITUT DE PROTECTION ET DE SÛRETÉ NUCLÉAIRE, FRANCE; NUCLEAR INSTALLATIONS INSPECTORATE, UNITED KINGDOM; NUCLEAR REGULATORY COMMISSION, UNITED STATES OF AMERICA, Four Party Regulatory Consensus Report on the Safety Case for Computer-Based Systems in Nuclear Power Plants, HMSO, Norwich (1997).

[8] INTERNATIONAL ATOMIC ENERGY AGENCY, Specification of Requirements for Upgrades Using Digital Instrument and Control Systems, IAEA-TECDOC-1066, Vienna (1999).

[9] COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE, Centrales nucléaires — Systèmes d'instrumentation et de contrôle-commande importants pour la sûreté — Classification, Norme n° 61226, CEI, Genève (1993).

[10] INTERNATIONAL NUCLEAR SAFETY ADVISORY GROUP, Basic Safety Principles for Nuclear Power Plants, Safety Series No. 75-INSAG-3 Rev. 1, INSAG-12, IAEA, Vienna (1999).

[11] INTERNATIONAL ATOMIC ENERGY AGENCY, Application of the Single Failure Criterion, Safety Series No. 50-P-1, IAEA, Vienna (1990).

[12] BRITISH COMPUTER SOCIETY, Guidelines on Good Security Practice, BCS, Swindon (1990).

[13] BRITISH STANDARDS INSTITUTION, Code of Practice for Information Security Management, BS 7799, BSI, London (1995).

[14] AGENCE INTERNATIONALE DE L'ÉNERGIE ATOMIQUE, L'assurance de la qualité pour la sûreté des centrales nucléaires et autres installations nucléaires: Code et guides de sûreté Q1-Q14, collection Sécurité n° 50-C/SG-Q, AIEA, Vienne (1999).

[15] INTERNATIONAL ATOMIC ENERGY AGENCY, Manual on Quality Assurance for Computer Software Related to the Safety of Nuclear Power Plants, Technical Reports Series No. 282, IAEA, Vienna (1988).

[16] INTERNATIONAL ATOMIC ENERGY AGENCY, Verification and Validation of Software Related to Nuclear Power Plant Instrumentation and Control, Technical Reports Series No. 384, IAEA, Vienna (1999).

[17] COMMISSION ÉLECTRONIQUE INTERNATIONALE, Calculateurs programmés importants pour la sûreté des centrales nucléaires, Norme n° 987, CEI, Genève (1989).

[18] VOGES, U., “Software diversity”, Proc. 9ème conférence annuelle sur la sûreté des logiciels, Luxembourg, 1992, Centre for Software Reliability, City Univ., Londres (1992).

[19] PAVEY, D.J., WINSBORROW, L.A., Demonstrating equivalence of source code and PROM contents, Computer J. 36 (1993) 654.

ANNEXE

UTILISATION ET VALIDATION D'UN LOGICIEL PRÉEXISTANT[1]

Cette annexe reproduit dans son intégralité (avec quelques modifications mineures) la section 1.3 de la réf. [A–1].

INTRODUCTION

A-1. L'incorporation de composants d'un logiciel préexistant (LPE) dans le logiciel d'application peut non seulement être avantageuse du point de vue de la productivité mais peut également accroître la sûreté d'un système logiciel si elle est effectuée de manière appropriée. L'avantage réside dans le fait que ces composants LPE ont souvent été utilisés dans de nombreuses applications et que l'expérience acquise lors de leur fonctionnement, lorsqu'elle peut être évaluée et est représentative, peut être prise en compte. Les composants logiciels réutilisables peuvent avoir été développés selon des normes suffisamment strictes dans d'autres secteurs industriels pour pouvoir être utilisés dans des applications essentielles pour la sûreté et, de ce fait, peuvent être réutilisables dans l'industrie nucléaire. Les exploitants peuvent avoir l'intention d'utiliser un tel logiciel à condition qu'une évaluation adéquate ait été effectuée.

PROBLÈMES INHÉRENTS

A-2. Le comportement fonctionnel et non fonctionnel (sûreté de fonctionnement, performance, …) des LPE n'est souvent ni clairement défini ni documenté.

A-3. La documentation et les données sur l'expérience d'exploitation du composant LPE ne sont souvent pas suffisamment appropriées pour fournir les preuves qui seraient nécessaires pour compenser le manque de connaissances sur le produit LPE et sur son processus de développement.

A-4. En conséquence des problèmes évoqués en A-2 et A-3, les critères d'acceptation et les procédures d'enquête servant à démontrer l'aptitude du composant LPE à remplir sa fonction pour une application spécifique peuvent être difficiles à mettre en place.

---

[1] © Communautés européennes. Reproduit avec la permission de l'éditeur, l'Office des publications officielles des Communautés Européennes.

A-5. Les antécédents opérationnels du composant LPE peuvent ne pas correspondre exactement à l'application voulue. De ce fait, l'application peut parfois utiliser des chemins d'accès du logiciel de qualité inconnue.

POSITION COMMUNE[2]

A-6. Les fonctions devant être remplies par les composants LPE doivent être clairement identifiées et l'impact sur la sûreté de ces fonctions doit être évalué.

A-7. Les composants LPE à utiliser doivent être clairement identifiés, y compris leur(s) version(s) de code.

A-8. Les interfaces au travers desquelles l'utilisateur ou d'autres logiciels appellent les modules LPE doivent être clairement identifiées et complètement validées. Il faut prouver qu'aucune autre séquence d'appel ne peut être exercée, même involontairement.

A-9. Les composants LPE doivent avoir été développés et doivent être tenus à jour conformément aux méthodes adéquates d'ingénierie logicielle et aux normes d'assurance de la qualité appropriées à leur utilisation prévue.

A-10. Pour les systèmes de sûreté, les composants LPE doivent être soumis à la même évaluation (analyse et revue) du produit final (pas celle du processus de production) que tout nouveau logiciel développé pour l'application. Si nécessaire, une méthode logique inverse doit être utilisée pour permettre la spécification complète du composant LPE à évaluer.

A-11. Si des modifications des composants LPE sont nécessaires, la documentation sur la conception et le code source de ces composants LPE doivent être disponibles.

A-12. Les informations nécessaires pour évaluer la qualité du produit LPE, de son analyse et des processus de développement doivent être disponibles ; ces informations doivent être suffisantes pour évaluer le composant LPE au niveau de qualité requis.

A-13. Pour l'acceptation, les actions suivantes doivent être prises :

[2] Dans le cadre de ce Guide de sûreté, ce terme signifie recommandations.

A-14. Vérifier que les fonctions remplies par le composant LPE satisfont à toutes les exigences exprimées dans les spécifications des exigences relatives au système de sûreté et dans les autres spécifications de logiciel applicables ;

A-15. Vérifier que les fonctions LPE non requises par les spécifications des exigences relatives au système de sûreté ne peuvent pas être appelées et compromettre les fonctions requises, par exemple par le biais d'entrées erronées, d'interruptions et de mauvaises utilisations ;

A-16. Effectuer une analyse de conformité de la conception du LPE par rapport aux exigences des normes applicables (par exemple [A–2]) ;

A-17. Les fonctions LPE que l'on désire utiliser doivent être validées à l'aide de tests. Les tests peuvent inclure des tests effectués par le fournisseur ;

A-18. S'assurer que les fonctions LPE ne peuvent pas être utilisées par le système de sûreté, par un autre logiciel ou par les utilisateurs d'une manière différente de celle qui a été spécifiée et testée (si nécessaire via l'implémentation de conditions préalables, de mécanismes de verrouillage ou d'autres protections).

A-19. Si on prend en compte le retour d'expérience pour le processus d'autorisation, on devra pouvoir disposer de suffisamment d'informations sur l'expérience d'exploitation et les taux de défaillances. Le retour d'expérience doit être correctement évalué sur la base d'une analyse de la durée de l'exploitation, des rapports d'erreur et de l'historique des versions des systèmes en exploitation. L'expérience d'exploitation doit également être basée sur l'utilisation des LPE dans des profils opérationnels identiques. Cette expérience en exploitation doit être basée sur la dernière version sauf si une analyse d'impact adéquate montre que l'expérience précédente basée sur des parties non modifiées du LPE est encore valable du fait que ces parties n'ont pas été affectées par des versions ultérieures.

A-20. Si les informations disponibles du type requis par la recommandation A-19 ci-dessus ne sont pas suffisantes, alors une analyse (évaluation des risques) de l'impact sur la sûreté d'une panne du composant LPE doit être effectuée. Une attention toute particulière doit être apportée aux effets de voisinage possibles et aux défaillances pouvant se produire au niveau des interfaces entre le composant LPE et l'utilisateur et/ou les autres composants du logiciel.

A-21. Les erreurs trouvées lors de la validation du composant LPE doivent être analysées et prises en compte dans la procédure d'acceptation.

PRATIQUES RECOMMANDÉES

A-22. L'expérience d'exploitation peut être considérée comme une preuve de type statistique complémentaire à la validation et à la vérification du logiciel du système (systèmes d'exploitation, protocoles de communication, fonctions standard [A–2, E3, approches statistiques]).

A-23. Les données d'évaluation du crédit que l'on peut accorder à l'expérience d'exploitation doivent être collectées en termes d'informations sur le site et de profils opérationnels, de taux de sollicitation et de durée d'exploitation, de rapports d'erreur et d'historique des versions.

Les informations sur le site et les données de profil opérationnel doivent comporter :

— la configuration du LPE ;
— les fonctions utilisées ;
— les types et caractéristiques des signaux d'entrée, y compris les domaines de valeurs et, si nécessaire, la cadence des modifications ;
— les interfaces utilisateur ;
— le nombre de systèmes.

Les données de taux de sollicitation et de durée d'exploitation doivent comporter :

— le temps écoulé depuis le premier démarrage ;
— le temps écoulé depuis la dernière version du LPE ;
— le temps écoulé depuis la dernière erreur grave (le cas échéant) ;
— le temps écoulé depuis le dernier rapport d'erreur (le cas échéant) ;
— le type et le nombre des sollicitations exercées sur le LPE.

Les rapports d'erreur doivent comporter :

— la date de l'erreur, la gravité ;
— les corrections apportées aux erreurs.

L'historique des versions doit comporter :

— la date et l'identification des versions ;
— les défauts corrigés, les modifications fonctionnelles ou les extensions ;
— les questions en suspens.

Ces données doivent être enregistrées avec l'identification de la version du composant LPE et de la configuration associée.

RÉFÉRENCES DE L'ANNEXE

[A–1] EUROPEAN COMMISSION, European Nuclear Regulators' Current Requirements and Practices for the Licensing of Safety Critical Software for Nuclear Regulators, Rep. EUR 18158 EN, Office for Official Publications of the European Communities, Luxembourg (1998).

[A-2] COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE, Logiciel pour les calculateurs utilisés dans les systèmes de sûreté des centrales nucléaires, Norme n° 880, CEI, Genève (1986).

[3] Dans le champ d'application de ce Guide de sûreté, ces définitions de validation et de vérification s'appliquent dans le contexte du cycle de vie d'un système programmé.

GLOSSAIRE

Les définitions suivantes s'appliquent aux fins de la présente publication.

architecture du système programmé. Les composants matériels (unités centrales, mémoires, périphériques d'entrée et de sortie) du système programmé, leurs connexions, les systèmes de communication et le mappage des fonctions logicielles sur ces composants.

cycle de vie du système. Tous les stades par lesquels passe un système, depuis sa conception jusqu'à sa mise au rebut finale.

exigences non fonctionnelles. Des propriétés ou performances devant être garanties.

exigences relatives à la sûreté fonctionnelle. Le service de sûreté ou la fonction de sûreté devant être fournis.

exigences relatives au logiciel. Instructions relatives au comportement que le composant du logiciel doit avoir pour satisfaire aux exigences relatives au système lorsque le logiciel s'exécute sur l'ensemble d'ordinateurs sélectionné avec les périphériques associés.

exigences relatives au système programmé. Les propriétés fonctionnelles et non fonctionnelles du système programmé nécessaires et suffisantes pour satisfaire aux exigences relatives à la sûreté de la centrale qui ont été définies à un niveau de conception supérieur.

implémentation. (1) Le processus consistant à convertir une conception en composants matériels ou en composants logiciels. (2) Le résultat du processus décrit en (1).

intégration du système. Le processus consistant à intégrer un système programmé aux autres composants d'un système d'une centrale.

intégration du système programmé. Le processus d'intégration du logiciel au matériel pour former le système programmé.

logiciel prédéveloppé ou logiciel préexistant. Un logiciel utilisé dans un système programmé qui n'a pas été développé sous le contrôle des responsables du développement du système (un logiciel de série du commerce est un type de logiciel préexistant).

logique de vote. Stratégie visant à réduire la probabilité d'une action intempestive du système en choisissant par un vote les signaux de sortie (comme le vote en deux sur trois).

méthode de rétro-ingénierie. La reconstruction d'une spécification à partir d'une conception terminée afin de faire la comparaison avec la spécification d'origine pour s'assurer que les deux sont compatibles.

micrologiciel. Des programmes informatiques et des données chargées dans une catégorie de mémoire (telle qu'une mémoire morte (ROM)) non modifiables par l'ordinateur au cours du traitement.

redondance. La mise en place de structures de remplacement (identiques ou différentes), systèmes ou composants, afin qu'un élément quelconque puisse remplir la fonction requise indépendamment de l'état de fonctionnement ou de défaillance d'un autre élément.

scrutabilité. Tout ce qui contribue à identifier des divergences par rapport aux résultats planifiés lors de l'évaluation des éléments du logiciel.

sûreté de fonctionnement. La fiabilité du service fourni permettant de faire confiance, à juste titre, à ce service. La fiabilité, la disponibilité et la sûreté sont des attributs de la sûreté de fonctionnement.

système de sûreté. Un système important pour la sûreté, permettant de garantir un arrêt sûr du réacteur ou l'évacuation de la chaleur résiduelle du cœur du réacteur ou pour limiter les conséquences d'accidents de référence ou de transitoires d'exploitation.

système important pour la sûreté. Un système qui fait partie des systèmes de sûreté et/ou dont le dysfonctionnement ou la panne peut conduire à une exposition aux rayonnements du personnel ou du public sur le site.

système lié à la sûreté. Un système important pour la sûreté qui ne fait pas partie d'un système de sûreté.

système programmé important pour la sûreté. Un système de la centrale important pour la sûreté dans lequel les fonctions de sûreté du système sont remplies par l'intermédiaire d'un système programmé intégré.

timing. Les limites pour le temps nécessaire aux transformations effectuées par un logiciel.

traçabilité. Le niveau de relation pouvant être établi entre deux produits du processus de développement, spécialement pour les produits ayant une relation prédécesseur–successeur entre eux.

validation[3]**.** Le processus consistant à tester et évaluer le système programmé intégré (matériel et logiciel) afin de garantir sa conformité par rapport aux exigences fonctionnelles, aux exigences relatives aux performances et à celles concernant les interfaces.

vérification[3]**.** Le processus consistant à garantir qu'une des phases du cycle de vie du système satisfait aux exigences qui lui sont imposées par la phase précédente.

[3] Dans le champ d'application de ce Guide de sûreté, ces définitions de validation et de vérification s'appliquent dans le contexte du cycle de vie d'un système programmé.

PERSONNES AYANT COLLABORÉ À LA RÉDACTION ET À L'EXAMEN

Asmis, G.J.K.	Commission de contrôle de l'énergie atomique (Canada)
Bafurík, J.	Autorité de sûreté nucléaire (Slovaquie)
Beltracchi, L.	Commission de la réglementation nucléaire (États-Unis)
Bouard, J.-P.	Électricité de France (France)
Carre, B.	Praxis Critical Systems plc (Royaume-Uni)
Chandra, U.	Centre de recherche atomique Bhabha (Inde)
Courtois, P.-J.	AV Nuclear (Belgique)
Duong, M.	Agence internationale de l'énergie atomique
Fandrich, J.	Siemens A.G (Allemagne)
Faya, A.	Commission canadienne de sûreté nucléaire (Canada)
Ficheux, F.	Électricité de France (France)
Geerinck, P.	TRACTEBEL S.A. (Belgique)
Greenberg, R.	Commission de l'énergie atomique (Israël)
Hamar, K.	Commission de l'énergie atomique (Hongrie)
Henry, J.Y.	Institut de protection et de sûreté nucléaire (France)
Hirose, M.	Nuclear Power Engineering Corporation (Japon)
Hohendorf, R.J.	Ontario Hydro (Canada)
Hughes, P.	Service d'inspection des installations nucléaires (Royaume-Uni)
Karpeta, C.	Autorité tchèque de sûreté nucléaire (République tchèque)

Kersken, M.	Institut für Sicherheitstechnologie GmbH (Allemagne)
Kulig, M.J.	Agence internationale de l'énergie atomique
Lawrence, D.	Lawrence Livermore National Laboratory (États-Unis)
Lee, J.-S.	Institut de recherche sur l'énergie atomique (République de Corée)
Mandij, D.	Centrale nucléaire de Krško (Slovénie)
Nechanický, M.	Centrale nucléaire de Temelin (République tchèque)
Pachner, J.	Agence internationale de l'énergie atomique
Regnier, P.	Institut de protection et de sûreté nucléaire (France)
Roca, J.L.	Ente Nacional Regulador Nuclear (Argentine)
Saidel, F.	Office fédéral de radioprotection (Allemagne)
Tanaka, T.	Tokyo Electric Power Company (Japon)
Taylor, R.P.	Commission canadienne de sûreté nucléaire (Canada)
Vojtech, J.	Institut de recherche sur les centrales nucléaires (Slovaquie)
Voumard, A.	Division principale de la sécurité des installations nucléaires (Suisse)
Wainwright, N.	Service d'inspection des installations nucléaires (Royaume-Uni)
Yates, R.L.	Service d'inspection des installations nucléaires (Royaume-Uni)
Zambardi, F.	Agence nationale pour la protection de l'environnement (Italie)

ORGANES CONSULTATIFS POUR L'APPROBATION DES NORMES DE SÛRETÉ

Comité consultatif pour les normes de sûreté nucléaire

Allemagne : Wendling, R.D., Sengewein, H., Krüger, W. ; *Belgique :* Govaerts, P. (Président) ; *Brésil :* da Silva, A.J.C. ; *Canada :* Wigfull, P. ; Chine : Lei, Y., Zhao, Y. ; *États-Unis d'Amérique :* Morris, B.M. ; *Fédération de Russie :* Baklushin, R.P. ; *Finlande :* Salminen, P. ; *France :* Saint Raimond, P. ; *Inde :* Venkat Raj, V. ; *Japon :* Tobioka, T. ; *Pays-Bas :* de Munk, P., Versteeg, J. ; *République de Corée :* Moon, P.S.H. ; *République tchèque :* Stuller, J. ; *Royaume-Uni :* Willby, C., Pape, R.P. ; *Suède :* Viktorsson, C., Jende, E. ; *Agence de l'OCDE pour l'énergie nucléaire :* Frescura, G., Royen, J ; *AIEA :* Lacey, D.J. (Coordonnateur).

Commission consultative pour les normes de sûreté

Allemagne : Hennenhöfer, G., Wendling, R.D. ; *Argentine :* Beninson, D. ; *Australie :* Lokan, K., Burns, P. ; *Canada :* Bishop, A. (Président), Duncan, R.M. ; *Chine :* Huang, Q., Zhao, C. ; *Espagne :* Alonso, A., Trueba, P. ; *États-Unis d'Amérique :* Travers, W.D., Callan, L.J., Taylor, J.M. ; *France :* Lacoste, A.-C., Asty, M. ; *Japon :* Sumita, K., Sato, K. ; *République de Corée :* Lim, Y.K. ; *Royaume-Uni :* Williams, L.G., Harbison, S.A. ; *Suède :* Holm, L.-E. ; *Suisse :* Prêtre, S. ; *Slovaquie :* Lipár, M., Misák, J. ; *Agence de l'OCDE pour l'énergie nucléaire :* Frescura, G. ; *AIEA :* Karbassioun, A. (Coordonnateur) ; *Commission internationale de protection radiologique :* Valentin, J.