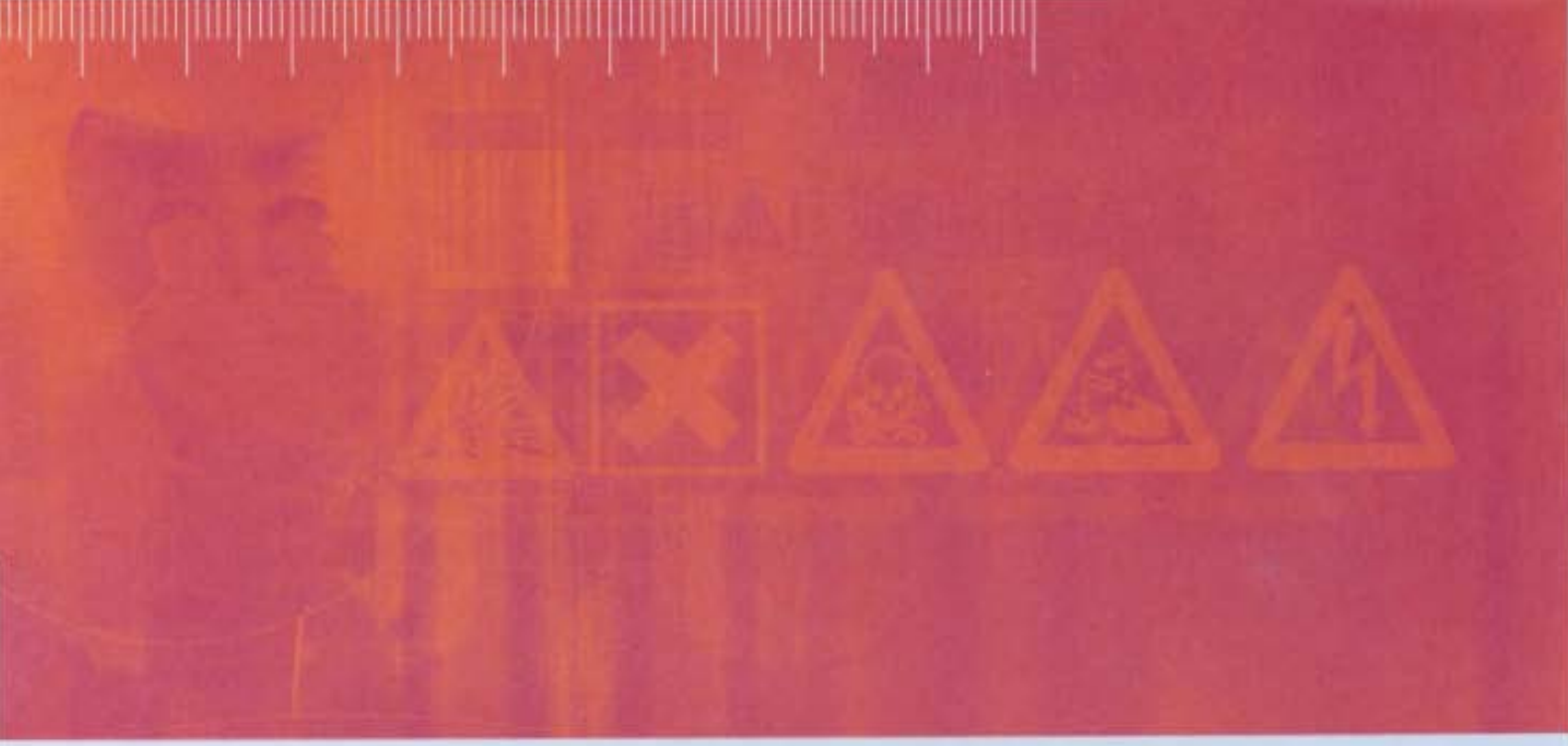


RAPPORT D'ÉTUDE 20/07/2016

N° DRA-14-142065-06403E

Guide d'inspection des logiciels applicatifs
Principes et mise en œuvre
VERSION FINALE

INERIS

maîtriser le risque
pour un développement durable

Guide d'inspection des logiciels applicatifs

Principes et mise en œuvre

Verneuil-en-Halatte
Direction des Risques Accidentels

Liste des personnes ayant participé à l'étude : François MASSÉ

PRÉAMBULE

Le présent rapport a été établi sur la base des informations fournies à l'INERIS, des données (scientifiques ou techniques) disponibles et objectives et de la réglementation en vigueur.

La responsabilité de l'INERIS ne pourra être engagée si les informations qui lui ont été communiquées sont incomplètes ou erronées.

Les avis, recommandations, préconisations ou équivalent qui seraient portés par l'INERIS dans le cadre des prestations qui lui sont confiées, peuvent aider à la prise de décision. Etant donné la mission qui incombe à l'INERIS de par son décret de création, l'INERIS n'intervient pas dans la prise de décision proprement dite. La responsabilité de l'INERIS ne peut donc se substituer à celle du décideur.

Le destinataire utilisera les résultats inclus dans le présent rapport intégralement ou sinon de manière objective. Son utilisation sous forme d'extraits ou de notes de synthèse sera faite sous la seule et entière responsabilité du destinataire. Il en est de même pour toute modification qui y serait apportée.

L'INERIS dégage toute responsabilité pour chaque utilisation du rapport en dehors de la destination de la prestation.

	Rédaction	Vérification	Approbation
NOM	François MASSÉ	Ahmed ADJADJ	Sylvain CHAUMETTE
Qualité	Ingénieur Unité QRIB Direction des Risques Accidentels	Responsable Programme Direction des Risques Accidentels	Responsable Pole AGIR Direction des Risques Accidentels
Visa			

TABLE DES MATIÈRES

1. INTRODUCTION ET OBJECTIF DU GUIDE

1.1 CONTEXTE DE L'ÉTUDE

L'évaluation des MMRI, suivant les doctrines et guides en usage, se fonde sur une décomposition en sous-systèmes « détection », « traitement » et « action » qui sont évalués séparément. Le niveau de confiance de chaque sous-système est évalué en fonction de l'architecture matérielle et de la technologie. Pour les sous-systèmes « traitement », et en particulier les automates programmables de sécurité, le niveau de confiance peut être évalué à partir de la certification SIL du matériel.

Dans cette démarche, l'automate est vu comme une boite noire et le logiciel applicatif est peu pris en compte dans l'évaluation du Niveau de Confiance. En effet, l'évaluation du développement et de la validation des logiciels nécessite d'analyser les barrières techniques de sécurité avec un niveau de détail rarement appliqué dans le cadre de l'appréciation de la maîtrise globale du risque d'une installation.

Or, le logiciel applicatif doit répondre aux exigences fonctionnelles et de performance de la MMRI : la bonne programmation des conditions de déclenchement et la logique de vote définissent l'efficacité et la performance de la barrière. Celles-ci peuvent donc être affectées par une mauvaise conception du logiciel. Ainsi, les modes d'exploitation particuliers ou les comportements sur défauts doivent être identifiés pour atteindre le Niveau de Confiance (NC) requis. De plus, des défauts de conception ou de maintenance du logiciel peuvent être à l'origine de défaillances systématiques qui rendent la barrière inefficace ou inopérante. Enfin, le logiciel applicatif peut être un mode commun entre plusieurs MMRI dont les traitements sont réalisés par les mêmes automates.

Le retour d'expérience sur les automates et l'état de l'art sur leur utilisation pour des fonctions critiques démontrent qu'il est nécessaire de traiter correctement les aspects logiciels. Ainsi :

- Le retour d'expérience de l'INERIS, sur la certification de fonctions de sécurité utilisant des automates programmables, montre que souvent :
    - les spécifications sont incomplètes ;
    - les tests de modes dégradés ou de comportement sur défaut mettent en évidence des défaillances systématiques de la fonction ;
    - la gestion des modifications présente des lacunes dans la validation et la mise en œuvre.
- Le BARPI a réalisé une étude sur les automatismes de conduite des procédés [1] qui identifie des accidents dont la cause ou l'aggravation sont dues à un défaut de la fonction traitement. Cette étude ne porte pas sur les automates de sécurité à proprement dit mais sur la fonction de conduite. Elle met néanmoins en évidence la difficulté pour un exploitant à prendre en compte avec un niveau de détail suffisant l'utilisation des automates et en particulier le développement du logiciel applicatif et la gestion de leur modification.

- La norme de sécurité fonctionnelle IEC 61511 [2] impose des exigences détaillées sur le développement des systèmes instrumentés de sécurité (SIS). Les exigences de cette norme portant sur l'architecture matérielle, les processus de validation, de tests et de maintenance périodique sont bien transposées dans la réglementation relative aux MMRI et les différents guides ou méthodes qui s'y rapportent [5], [6], [7]. En revanche, les exigences portant sur le logiciel applicatif y sont très peu prises en compte.

Il apparait donc nécessaire de fournir des éléments permettant de prendre en compte les logiciels applicatifs dans l'évaluation des MMRI. L'évaluation doit être adaptée aux problématiques des exploitants et des inspecteurs des installations classées.

1.2 OBJECTIFS

Ce document a pour objectif de présenter :

- un état de l'art très général sur le développement et la validation des logiciels applicatifs des automates programmables dans le cadre des MMRI ;
- des critères d'inspection spécifiques aux logiciels applicatifs des MMRI et une démarche d'inspection permettant de vérifier ces critères.

Le corps de ce document apporte les connaissances nécessaires à l'application de deux fiches jointes en annexes :

- Une fiche détaillée qui peut être utilisée par un exploitant comme support pour la mise en œuvre du cycle de vie logiciel et l'évaluation des logiciels applicatifs de sécurité (annexe 3) ;
- Une fiche d'inspection destinée aux Inspecteurs des Installations Classées pour évaluer la maîtrise des logiciels applicatifs (annexe 4).

Ces fiches présentent des éléments pour la conception et l'évaluation des logiciels applicatifs intégrés aux systèmes instrumentés de sécurité (SIS) ou pour les fonctions de sécurité réalisées par les automates de contrôle-commande de l'installation. Un SIS peut intégrer une ou plusieurs fonctions instrumentées de sécurité (SIF) qui peuvent être classées comme MMRI dans le cadre de l'étude de danger d'une ICPE. Un automate de contrôle-commande peut intégrer au maximum une fonction de sécurité par scénario d'accident qui pourra être classée comme MMRIc.

Les deux fiches couvrent le cycle de vie complet du logiciel. La fiche détaillée permet de planifier et de valider l'application des mesures techniques nécessaires dans les différentes phases de conception, validation et maintenance du logiciel. La fiche d'inspection permet de vérifier si la spécification et la validation du logiciel et si l'organisation mise en œuvre respectent les règles essentielles pour la maîtrise de logiciels applicatifs critiques.

1.3 LIMITES

Les fiches traitent des logiciels applicatifs des Automates Programmables Industriels (API) qu'ils soient de sécurité ou non. Les logiciels associés aux systèmes d'exploitation des automates (OS, firmware) n'entrent pas dans le cadre de cette étude.

Les éléments présentés sont spécifiques aux logiciels de sécurité mais ne sont pas adaptés aux logiciels de contrôle commande ou aux logiciels de supervision de l'installation.

Ces fiches ne permettent pas d'évaluer le niveau de confiance ou l'architecture d'une SIF complète ou d'un automate mais de vérifier que le processus de développement et de validation du logiciel applicatif, ainsi que les processus de maintenance sont compatibles avec le niveau de confiance requis pour la MMRI.

L'objectif de l'inspection est de vérifier que le logiciel applicatif est apte à réaliser sa fonction de sécurité avec le Niveau de Confiance requis. L'analyse des vulnérabilités de l'installation et sa protection contre la malveillance ne sont pas traitées dans ce document. Ces analyses ne peuvent pas se limiter aux fonctions de sécurité des logiciels applicatifs des automates mais nécessitent également d'évaluer les logiciels des systèmes d'exploitation de l'installation, les fonctions de contrôle-commande, les réseaux de communication, le système d'information du site, etc.

La démarche d'inspection ne nécessite pas de lire et d'analyser le codage du logiciel. Ces analyses sont difficilement réalisables dans le cadre d'une inspection car elles demandent du temps et des connaissances techniques très spécifiques. On s'assurera que l'exploitant a prévu l'application de règles de conception et que les activités de validation nécessaire sont mises en œuvre.

Ce guide se veut pragmatique, simple et utilisable par des non spécialistes. Il peut donc comporter quelques simplifications par rapport aux pratiques des spécialistes du sujet.

1.4 CONTENU DU DOCUMENT

Ce document présente, au chapitre 2, les technologies visées et les principes généraux pour l'évaluation des logiciels applicatifs. Ce chapitre rappelle également les critères d'évaluation des MMRI, leur lien avec les propriétés d'un logiciel sûr et définit un cycle de vie du logiciel autour de 5 thèmes :

- Le contexte technique et l'organisation ;
- les spécifications fonctionnelles du logiciel qui correspondent à la conception du point de vue de l'utilisateur ;
- La réalisation du logiciel qui est la conception du point de vue du fournisseur ;
- Les tests et la validation dont les responsabilités sont partagées entre utilisateur et concepteur du logiciel ;
- l'exploitation qui doit permettre de maintenir les performances au travers de la gestion des versions et des modifications et des tests périodiques.

Ces thèmes sont définis en fonction du retour d'expérience de l'INERIS sur l'évaluation des logiciels et des principaux référentiels normatifs ou méthodologiques [2][3][4]. Ils doivent également permettre de faire le lien avec les documents existants :

- la doctrine MMRI [6] ;
- le guide Vieillissement [7].

Ces cinq thèmes doivent permettre de réaliser une inspection couvrant les aspects fonctionnels, la maîtrise des exigences techniques et les processus supports. Le chapitre 3 présente de manière approfondie les objectifs recherchés et contenus détaillés de chacun des 5 thèmes.

Les critères d'inspection et sa mise en œuvre pratique sont présentés au chapitre 4.

La fiche détaillée et la fiche d'inspection en annexes 3 et 4 de ce document reprennent les éléments présentés au chapitre 3 et 4 en précisant s'ils sont requis pour des niveaux de confiance NC1, NC2 et/ou NC3.

2. PROCESSUS, CRITÈRES ET PRINCIPES DE CONCEPTION ET D'ÉVALUATION

2.1 TECHNOLOGIES ET ORGANISATIONS ÉVALUÉES

2.1.1 QU'EST-CE QUE LE LOGICIEL APPLICATIF ?

Le fonctionnement d'un automate programmable nécessite deux types de logiciels : le logiciel système et le logiciel applicatif.

Le **logiciel système** ou intégré fait partie de l'équipement fourni par le constructeur de l'automate et n'est pas accessible pour des modifications par l'utilisateur final. Il peut être désigné par les termes de firmware ou d'OS (Operating System[1]). C'est une couche logicielle qui offre une interface de programmation à l'utilisateur en le déchargeant de toutes les tâches nécessaires à l'exploitation des ressources matérielles. La machine physique constituée du processeur, des interfaces, des mémoires, etc., est gérée par l'Operating System. Celui-ci est donc un gestionnaire de ressources qui permet à l'opérateur d'écrire un programme, contenant une suite d'instructions, sans connaissance détaillée des différents constituants du système.

Le **logiciel applicatif**, ou programme d'application, est le logiciel traduisant les fonctions souhaitées par l'utilisateur final. Il contient des séquences logiques, des autorisations, des limites et des expressions qui contrôlent les entrées, les sorties, les calculs et les décisions nécessaires pour réaliser les fonctions instrumentées de sécurité.

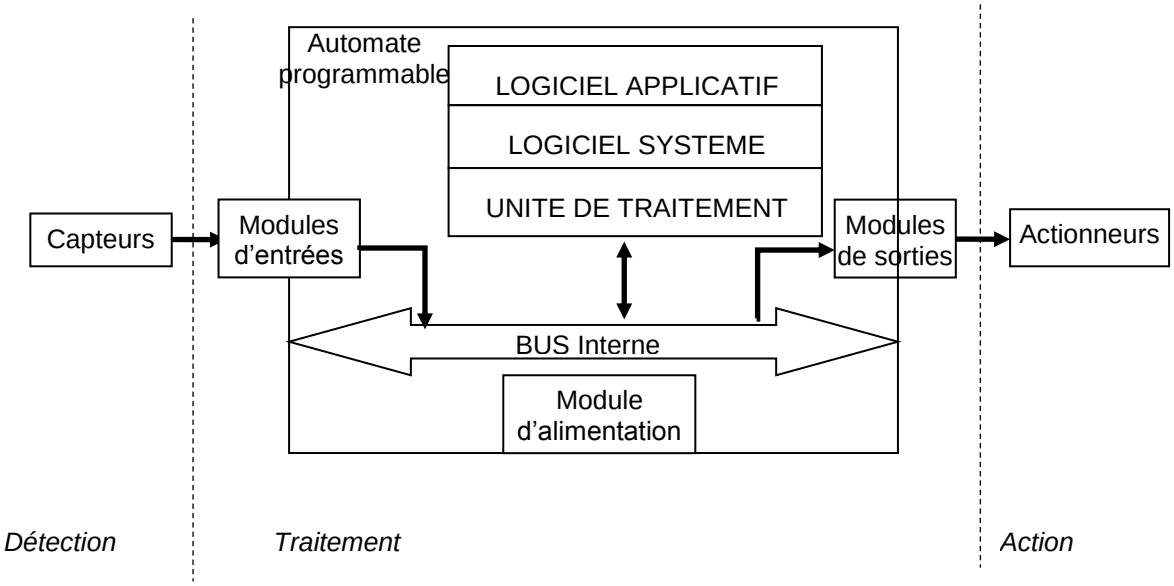


Figure 1 : Le logiciel applicatif dans l'automate et la MMRI

[1] Operating System : Système d'exploitation

Ce document porte sur l'évaluation des logiciels applicatifs qui est de la responsabilité de l'utilisateur de l'équipement. L'utilisateur doit cependant prendre en compte les interfaces entre logiciel applicatif et logiciel système. En effet, le logiciel applicatif est généralement développé et validé pour une version du logiciel système. Celui-ci est susceptible d'être mis à jour ce qui peut avoir un impact sur le bon fonctionnement de l'applicatif. L'utilisateur doit veiller au maintien des performances des fonctions de sécurité lors des modifications du logiciel système, si possible au moyen de tests.

2.1.2 DANS QUELS LANGAGES SONT-ILS DÉVELOPPÉS ?

La norme IEC 61511 définit trois types de langages de programmation qui peuvent être utilisés dans le développement des logiciels applicatifs :

- Les langages de programme figé (FPL) : l'utilisateur est limité au réglage de quelques jeux de paramètres prédéfinis et figés. L'utilisation de FPL correspond au paramétrage d'un appareil (par exemple le réglage de la gamme de mesure d'un transmetteur, d'un seuil d'alarme, ou le paramétrage d'une centrale feu et gaz).
- Les langages de variabilité limitée (LVL) sont conçus pour être compréhensibles par les utilisateurs et fournissent la possibilité de combiner des fonctions de bibliothèques spécifiques à une application. Il s'agit des langages utilisés pour la programmation des automates. Ils sont présentés en annexe 1.
- Les langages de variabilité totale (FVL) sont issus des systèmes d'information ou de l'informatique embarquée. Ils ne sont pas conçus pour respecter le fonctionnement synchrone des automates mais permettent de réaliser une gamme beaucoup plus étendue d'application pour l'électronique embarquée de manière générale. Il s'agit de langage tels que C, C#, Ada, Java, etc. Il est recommandé d'appliquer la norme IEC 61508-3 pour le développement d'application à partir de ces langages.

Les fiches d'évaluation annexées à ce document (annexes 3 et 4) sont adaptées aux logiciels applicatifs développés à partir de langages de variabilité limitée (LVL).

Ce document est limité à des évaluations de niveau NC3 au maximum.

Les fiches sont également applicables aux développements à partir de langages à variabilité figée (FPL), à l'exception de la fiche 3 sur la conception détaillée. Il est recommandé pour ces technologies de limiter l'évaluation à la spécification fonctionnelle, aux tests de validation, à la gestion des modifications et aux tests périodiques ce qui peut être fait avec la fiche simplifiée.

La démarche proposée pas les fiches détaillées et fiches d'inspection est insuffisante pour des langages FVL. Pour l'évaluation de logiciels SIL4 ou utilisant des langages à variabilité totale (FVL), il est recommandé de faire certifier par un tiers la conformité aux normes IEC 61508-3.

2.1.3 A QUELS ÉQUIPEMENTS SONT-ILS INTÉGRÉS ?

Ce document traite des logiciels applicatifs destinés à réaliser des fonctions de sécurité. Ces logiciels peuvent être intégrés à des automates de sécurité (souvent certifiés), ou à des automates de contrôle commande standards[2] (c'est-à-dire non conçus a priori pour réaliser des fonctions critiques). La programmation des automates utilise des langages à variabilité limitée.

Les centrales feu et gaz permettent en général de réaliser des paramétrages limités : réglage des seuils d'alarme, logique de vote entre plusieurs entrées, affectation de relais de sorties, temporisations, gestion de défauts, modes d'acquittement. Ce paramétrage peut être considéré comme un logiciel applicatif et pourra être vérifié au moyen des grilles d'évaluation. Le paramétrage des centrales peut être assimilé à un langage à variabilité figée.

Les logiciels systèmes de l'automate, les logiciels embarqués sur les différents capteurs ou actionneurs, les logiques à relayage, les blocs logiques de sécurité ne rentrent pas dans le cadre de ces fiches d'évaluation.

2.1.4 COMMENT SONT-ILS DÉVELOPPÉS ?

La conception d'un système logiciel met en œuvre un ensemble d'activités qui, à partir de la définition d'une fonction instrumentée, permet le développement, l'intégration et la validation d'un logiciel dans un environnement spécifié.

Le processus regroupant ces différentes activités est le cycle de vie du logiciel. Différents modèles peuvent décrire le cycle de vie d'un logiciel. Pour les logiciels de sécurité, on utilise généralement le modèle du cycle en V décrit ci-dessous.

Ce cycle permet de faire correspondre les étapes de conception aux étapes de vérification et de validation. La branche descendante correspond aux étapes de spécification et de réalisation, partant des exigences de haut niveau et aboutissant à une conception détaillée. La branche ascendante correspond aux différentes étapes d'intégration du logiciel dans son environnement et aux tests correspondants. Chaque phase de test est associée à une activité de spécification ou de conception. Par exemple, il est contrôlé que le système possède bien les fonctionnalités attendues (Spécification système) lors de la phase de validation.

[2] La doctrine MMRI autorise en effet à valoriser une fonction de sécurité réalisée par système de conduite de l'installation. Cette fonction de sécurité aura un facteur de réduction de risque limité à 10. La norme IEC 61511 permet sous certaines conditions de valoriser deux fonctions de ce type.

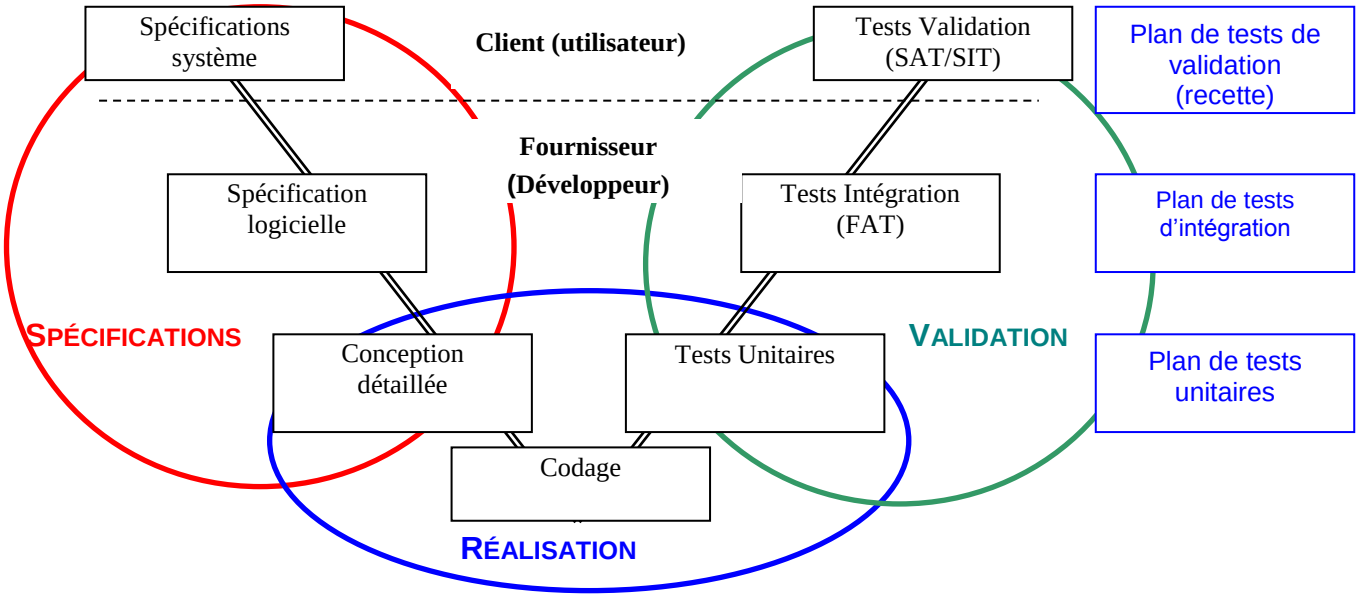


Figure 2 : Cycle de développement en V

Il est courant, dans les processus industriels de simplifier ce cycle, cependant, des phases minimales doivent être respectées :

- Une phase de spécification fonctionnelle :

Cette phase comprend l'étudie des fonctions à remplir par le système et en particulier les spécifications, la conception préliminaire et la conception détaillée.

- Une phase de réalisation :

Cette phase comprend une spécification détaillée, le développement et les tests unitaires des différents modules du logiciel.

- Une phase de validation :

Cette phase comprend l'intégration du logiciel dans son environnement d'utilisation et sa validation. Pour les procédés industriels, les différentes activités de validation peuvent être identifiées en tant que FAT, SAT et SIT.

- Une Phase d'exploitation :

Suite à la mise en service d'un logiciel validé, il faut s'assurer que ses performances sont maintenues, qu'il reste conforme aux spécifications et que des défauts éventuellement détectés sont corrigés.

2.1.5 QUI INTERVIENT SUR LES LOGICIELS APPLICATIFS ?

L'organisation du développement du logiciel applicatif a des implications sur la fiabilité des fonctions de sécurité et sur sa démonstration. La fiabilité du logiciel repose sur la maîtrise de son développement, de sa validation et de ses modifications.

La réalisation, la validation et les modifications des composantes logicielles des fonctions peuvent être de la responsabilité de l'utilisateur et/ou d'un ou plusieurs fournisseurs. La justification de la compétence des différents intervenants et la sensibilisation aux exigences spécifiques des différentes fonctions de sécurité doivent être vérifiées.

L'exploitant est responsable de la validation de la fonction de sécurité dans son ensemble ; il doit donc s'assurer, d'un point de vue technique ou contractuel que le développement est conforme aux exigences du niveau de confiance requis et que les différents intervenants sont compétents. Des processus qualité doivent identifier le cycle de vie, les responsabilités et les compétences des différents intervenants.

En règle générale, les phases initiales et finales du cycle de vie qui ont trait aux spécifications et à la validation de haut niveau sont de la responsabilité du client (utilisateur). Les phases de spécification et de conception détaillées sont réalisées par un fournisseur (développeur). Le fournisseur peut être interne ou externe à l'entreprise du client.

Lorsque le logiciel est en service, des processus supports doivent permettre de gérer les versions et les modifications en maintenant le niveau de sécurité.

2.2 CRITÈRES ET PRINCIPES D'ÉVALUATION

2.2.1 RAPPEL SUR LES CRITÈRES D'ÉVALUATION D'UNE MMR

Les critères d'évaluation d'une MMR, tels que définis dans la méthode Ω10 sont :

- l'indépendance par rapport à l'évènement initiateur ;
- l'efficacité ;
- le temps de réponse ;
- le niveau de confiance ;
- le maintien des performances dans le temps.

Les critères de performances sont définis d'une manière globale pour une fonction de sécurité réalisée en partie par des composants logiciels. Les critères de performances doivent donc être intégrés aux exigences du logiciel applicatif. Le cycle de vie du logiciel doit être défini avec l'objectif de répondre à ces critères.

2.2.2 CRITÈRES DE PERFORMANCE D'UN PROGRAMME APPLICATIF SÛR

Les notions de sûreté de fonctionnement des logiciels sont définies par plusieurs critères spécifiques. Parmi ceux-ci, les suivants sont ceux qui permettent de répondre aux objectifs d'évaluation des MMRI :

La Fiabilité : La fiabilité représente la performance prévisible et constante du logiciel dans les conditions spécifiées dans les exigences. Cet attribut est important pour la sécurité, car il réduit la probabilité que des pannes susceptibles de provoquer des dysfonctionnements soient introduites dans le programme d'application lors de sa mise en œuvre. La fiabilité s'obtient sur l'ensemble du cycle de vie du logiciel.

La Robustesse : La robustesse est la capacité du logiciel à fonctionner de manière acceptable en cas d'événements ou de conditions anormaux. Cet attribut est important pour la sécurité, car il représente la capacité du logiciel à prendre en compte les conditions hors limites, à être tolérant aux défaillances matérielles et à éviter la propagation d'erreurs de traitement ou de programmation. La robustesse est le résultat de bonnes spécifications et de bonnes pratiques de programmation

La Traçabilité : La traçabilité fait référence à la gestion des versions. L'ensemble des documents de développement et de validation doivent permettre de démontrer que la version logicielle implémentée répond aux performances et aux exigences attendues. L'ensemble des outils et bibliothèques utilisés doivent être identifiés. La traçabilité inclut également la capacité à associer le programme d'application à des documents de conception de niveau supérieur. Cet attribut est important pour la sécurité, car il facilite la vérification et la validation du logiciel. La traçabilité s'obtient par la réalisation de plans de tests couvrant l'ensemble des exigences puis par les gestions des versions et des modifications.

La Maintenabilité : La maintenabilité traduit la capacité du logiciel à être maintenu ou modifié après sa mise en service sans que des pannes soient introduites. Cet attribut est important pour la sécurité, car il réduit la probabilité d'une mauvaise exploitation suite à la présence de pannes lors de la maintenance du logiciel et permet de maintenir les performances dans le temps. La maintenabilité s'obtient par la mise en place de bonnes pratiques de conception détaillée.

Le lien entre les critères de performance d'un logiciel applicatif et les critères de performance de la MMRI ainsi que les thèmes d'inspection permettant de vérifier ces critères sont donnés dans le tableau ci-dessous :

Critère MMRI	Propriété du logiciel
Indépendance	Robustesse Traçabilité
Efficacité	Robustesse Traçabilité Fiabilité
Temps de réponse	Fiabilité Traçabilité Robustesse
Niveau de confiance	Fiabilité Robustesse Traçabilité
Maintien dans le temps	Traçabilité Maintenabilité

Tableau 1 : Critères de performance des logiciels au regard des critères de performance des barrières

2.2.3 DÉMARCHE DE CONCEPTION ET D'ÉVALUATION

Nous avons vu qu'il était possible de décrire le cycle de vie d'un logiciel en 3 phases de spécification, conception et validation auxquelles s'ajoute la maîtrise des logiciels en service et différents processus supports.

Pour planifier la conception ou valider un logiciel, l'exploitant a accès aux personnes ou entités responsables des différentes phases et processus de son

cycle de vie. L'évaluation doit donc entrer dans le détail des rôles et responsabilités des intervenants dans les différentes phases de ce cycle. Pour cela, les grilles détaillées sont structurées selon 5 thèmes, le premier ayant trait à la mise en œuvre de l'organisation et les 4 suivants au déroulement chronologique du cycle de vie :

Thème 1 : Contexte et organisation

Thème 2 : Spécifications fonctionnelles

Thème 3 : Conception détaillée

Thème 4 : Tests et validation

Thème 5 : Suivi du logiciel en exploitation

Ces thèmes sont développés au chapitre 4. Les fiches détaillées listent l'ensemble des exigences ou des vérifications applicables pour chacun d'entre eux.

En revanche, on considère que le rôle de l'Inspecteur des Installations Classées n'est pas d'interroger et d'évaluer les fournisseurs du logiciel applicatif, qui peuvent bien souvent être des sous-traitants, mais d'évaluer ce qui relève de la responsabilité de l'exploitant de l'installation.

L'évaluation doit permettre de vérifier les critères de performance des MMRI, pour leur partie logicielle. Pour cela, on évalue si les moyens et techniques mis en œuvre à chaque phase du cycle de vie permettent de développer et maintenir un logiciel sûr.

L'inspection suivra donc le cycle de vie du logiciel applicatif de manière chronologique et débutera par la spécification fonctionnelle qui permet de faire le lien avec l'évaluation des MMRI. L'objectif de l'inspection sera d'analyser la maitrise des logiciels applicatifs au travers de 5 questions :

- Q1 : Le développement du logiciel est-il issu d'une spécification fonctionnelle détaillée de la MMRI ?
- Q2 : De bonnes pratiques de programmation sont-elles appliquées ?
- Q3 : Un plan de validation permet-il de vérifier l'ensemble des exigences ?
- Q4 : S'assure-t-on que les performances du logiciel sont maintenues dans le temps ?
- Q5 : Les processus relatifs au logiciel permettent-ils d'atteindre et maintenir un niveau de sécurité ?

Les parties « spécification » et « validation » qui sont regroupées dans le même thème du cycle de vie (thèmes 2) se réfèrent à deux questions différentes (Q1 et Q3).

La fiche d'inspection donne des éléments précis pour répondre à ces 5 questions. Elle demande une analyse moins approfondie que la fiche détaillée, notamment pour la question 2 portant sur les techniques de développement du logiciel à proprement dit.

3. CONTENU DES THÈMES DU CYCLE DE VIE

Ce chapitre présente le contenu des 4 thèmes associés au cycle de vie. Les différents points abordés sont repris sous forme de grilles détaillées (Annexe 3).

La fiche d'évaluation détaillée reprend l'ensemble des points abordés dans ce chapitre et les présente sous forme de 4 tableaux correspondants aux 4 thèmes.

La fiche d'inspection simplifiée (annexe 4) reprend les points principaux de ce chapitre pour apporter des éléments de réponses aux 5 questions définies au chapitre 2.

3.1 THÈME 1 : CONTEXTE ET ORGANISATION

3.1.1 CONTEXTE TECHNIQUE : LES MMRI SUR L'INSTALLATION

Un exploitant doit être en mesure de lister les MMRI de son installation, la fonction de sécurité associée et le Facteur de Réduction de Risque ou Niveau de confiance requis.

Outre ces éléments fonctionnels, il doit avoir la maîtrise des informations sur les technologies utilisées afin de définir les exigences ou référentiels applicables : les MMRI peuvent être implémentées sur des automates de sécurité, des centrales feu et gaz ou des automates de contrôle commande ; un automate de sécurité peut exécuter plusieurs fonctions de sécurité qui peuvent intervenir sur le même scénario d'accident ou sur des scénarios différents.

L'âge du logiciel est également une information pertinente : certains logiciels peuvent être développés depuis de nombreuses années et, dans ce cas, il manquera certainement des documents de conception ou de développement pour justifier leur niveau de confiance L'ancienneté de la technologie peut également être source de perte de compétence ce qui complexifie la maintenance et les modifications du logiciel. En revanche, un logiciel en service depuis de nombreuses années sans modifications peut être considéré fiable si des tests fonctionnels ont été réalisés régulièrement ou si du retour d'expérience a été enregistré.

Enfin, un retour d'expérience sur les MMRI peut faire apparaître des informations sur les logiciels applicatifs : les sollicitations des fonctions de sécurité, les déclenchements intempestifs et l'historique des modifications.

Les informations recherchées lors de cette étape sont[3] :

Informations sur le contexte	
Présentation des MMRI de l'installation	- Fonctions de sécurité et NC - Automates utilisés, types (automates de sécurité, automates de contrôle commande, centrales feu et gaz)
Référentiels appliqués	- Normes appliquées pour le développement ou la certification des logiciels applicatifs
Age du logiciel	- Antériorité du logiciel (ancien projet peu documenté, projet récent…)
Retour d'expérience sur les MMRI	- Retour d'expérience sur la sollicitation des MMRI - Déclenchements intempestifs des MMRI - Dans quelles circonstances ? - Est-ce tracé ?
Exemple de documents	
➢ Liste des MMRI, APR, HAZOP, nœuds papillons ➢ Certification des MMRI	

Tableau 2 : Informations sur le contexte technique

3.1.2 L'ORGANISATION ET LES RESPONSABILITÉS POUR LES LOGICIELS APPLICATIFS

L'organisation mise en place pour le développement, la validation et la maîtrise des SIS doit être définie. Cette organisation doit en particulier faire apparaitre les personnes, services, organisations qui sont responsables de l'exécution et de la vérification de chacune des phases du cycle de vie.

Le cycle de vie doit au moins faire apparaitre :

- **la spécification d'exigences générales pour la fonction ;**
- **la spécification d'exigences pour le logiciel applicatif ;**
- **la réalisation du logiciel applicatif ;**
- **la validation du logiciel applicatif (qui peut inclure des tests en usine (FAT) et doit inclure des tests sur site (SAT)) ;**
- **la maintenance et les modifications du logiciel applicatif.**

Différentes entités ou sous-traitants peuvent être responsables selon les étapes, notamment pour la réalisation du logiciel. Certaines responsabilités sont toujours du ressort de l'utilisateur :

[3] La structure documentaire devrait en théorie justifier la conformité à l'ensemble des exigences des référentiels normatifs appliqués pour le développement ou la certification des logiciels. Parmi les référentiels on peut citer les normes de sécurité fonctionnelle IEC 61508 et IEC 61511 ou la norme directive Machines NF EN 13849 mais également des normes d'assurance qualité génériques (ISO 9001) ou spécifiques au logiciel (ISO 12207) et la norme relative à la programmation des automates IEC 61131-3.

- définir l'organisation mise en œuvre pour atteindre la sécurité et les responsabilités ;
- vérifier que les personnes impliquées sont informées de leur responsabilité et compétentes pour leur mission : connaissance des technologies qu'elles doivent mettre en œuvre et des exigences de sûreté applicables ;
- vérifier que l'indépendance entre les personnes chargées de la réalisation et les personnes chargées de la validation est suffisante[4] ;
- valider la fonction de sécurité.

Les critères retenus dans les grilles d'évaluation sont les suivants :

Exigences	
Cycle de vie appliqué	- Description du cycle de vie et des différentes étapes de spécification, de développement et de validation du logiciel
Responsabilités et compétences pour le développement des logiciels	- Entités ou personnes responsables de la maîtrise de différentes phases du cycle de vie (spécification, réalisation, validation, maintenance, modification) - Personnes et entités impliquées dans la réalisation des différentes phases (sous-traitance, compétence interne) - Compétences en sécurité fonctionnelle - Compétence sur les technologies mises en œuvre (équipements et outils de programmation utilisés)

[4] Le recours à une personne indépendante de l'équipe de projet a pour but d'accroître l'objectivité de l'évaluation. Le recours à une personne de haut niveau, (par exemple, justifié par : l'expérience, la formation, la position) a pour but d'assurer que les préoccupations exprimées par l'équipe de projet seront dûment prises en compte et traitées.

En fonction de l'organisation et de l'expertise au sein de l'entreprise, la sollicitation d'un organisme externe peut être nécessaire pour répondre à l'exigence d'un examinateur indépendant. Inversement, les entreprises disposant d'instances internes compétentes dans les domaines de l'évaluation des risques et de l'application des systèmes instrumentés de sécurité, indépendantes et distinctes (de par leur gestion et autres ressources) des instances responsables du projet, peuvent avoir la possibilité d'utiliser leurs propres ressources pour satisfaire à cette exigence d'un organisme indépendant.

Exigences	
Indépendance entre les responsables des différentes phases	Les personnes qui réalisent les FAT et les SAT doivent être indépendantes des personnes qui réalisent le développement. Au minimum, les niveaux d'indépendance suivants sont requis : - pour un niveau NC1, une personne indépendante réalise les tests ; - pour un NC2, un service indépendant est chargé des tests ; - pour un NC3 une organisation indépendante (organisme tiers) valide les tests. Le responsable de l'acceptation globale doit être l'utilisateur
Exemple de documents	
➢ Processus ou plan qualité relatif aux logiciels applicatifs ou à un projet ➢ Cahier des charges pour la sous-traitance ➢ Grilles de compétence	

Tableau 3 : Principales exigences organisationnelles

3.2 THÈME 2 : LES SPÉCIFICATIONS FONCTIONNELLES

3.2.1 LES SPÉCIFICATIONS FONCTIONNELLES

Les spécifications fonctionnelles décrivent le comportement attendu de la MMRI de manière générale. Elles doivent s'attacher à décrire le scénario envisagé et les principes de fonctionnement en prévention et / ou protection.

Ces spécifications correspondent à la SRS[5] telle que définie dans la norme IEC 61511. Elles doivent résulter des exigences identifiées pour les fonctions de sécurité lors de l'analyse des risques.

Si plusieurs MMRI interviennent sur le même scénario d'accident, des exigences d'indépendance doivent être spécifiées.

Une architecture matérielle générale doit être définie afin d'atteindre le niveau de confiance requis. Les choix d'architecture doivent prendre en compte la tolérance aux défaillances pour les capteurs et actionneurs (redondance), le choix technologique pour le ou les automates (le traitement est-il réalisé par un automate de contrôle commande, un automate de sécurité, plusieurs automates en redondance ?).

La spécification fonctionnelle servira de donnée d'entrée à la spécification logicielle et à sa réalisation.

Les spécifications fonctionnelles sont de la responsabilité de l'exploitant. Elles peuvent être présentées dans un cahier des charges destiné au développement du logiciel applicatif.

Ces spécifications doivent être testables et être couvertes par le plan de test de validation.

Les éléments devant apparaitre dans les spécifications fonctionnelles sont les suivants :

Exigences	
Démarche de conception	**Critères d'inspection**
Descriptions de la fonction de sécurité et du scénario envisagé	Définition de la MMR et de son fonctionnement global
Identification des différents MMRI intervenant sur le même scénario	L'évaluation des barrières intervenant sur un même scénario doit faire apparaitre s'il s'agit de barrières instrumentées afin d'évaluer leur indépendance
Description de l'architecture matérielle	Identification des capteurs et des points de mesure Automates utilisés, types et architectures Identification des actionneurs

[5] SRS : Safety Requirement Specification – Spécification des exigences de sécurité

Exigences	
Démarche de conception	**Critères d'inspection**
Conditions et seuils de déclenchement de l'action de sécurité	Logiques de vote entre les capteurs Les seuils doivent être indiqués dans la spécification fonctionnelle. Ils doivent être compatibles avec la cinétique du scénario.
Définition de l'action de sécurité	Séquence de commande des actionneurs, temporisations éventuelles Type de commande des actionneurs (TOR à manque ou à émission, numérique) Identification des alarmes
Comportements sur défauts	Comportements attendus sur défauts des capteurs, actionneurs et automate Les signaux d'alarme des capteurs et les valeurs hors échelle doivent être traités (alarme pour mise en place de moyens compensatoires ? déclenchement ? etc.), pour des signaux d'entrée redondants il peut y avoir des tests de cohérence Quels comportements sont attendus sur ces défauts ? de même, des moyens de diagnostics peuvent être intégrés aux actionneurs, comment sont-ils traités (ex pour les vannes : positionneurs, fin de course ou test de course partielle)
Exigences de temps de réponse	Temps de réponse de la MMR globale et temps de réponse attendu pour le capteur
Spécification des IHM	Si des actions humaines sont prévues, les moyens d'alarmes et d'affichage doivent être identifiés. Ils doivent être visibles et non ambigus, les comportements sur défauts doivent être spécifiés
Testabilité des exigences	Un plan de test de la MMRI (Plan de SAT) doit permettre de vérifier que l'ensemble des spécifications est correctement implémenté
Exemple de documents	
➢ SRS ou spécification ➢ Cahier des charges pour la sous-traitance ➢ Plan de test d'acceptation (ou plan de SAT)	

Tableau 4 : Principales spécifications de la fonction de sécurité exploitées dans le développement du logiciel applicatif

3.3 THÈME 3 : LA CONCEPTION DÉTAILLÉE DU LOGICIEL APPLICATIF

La phase de conception détaillée comprend la spécification détaillée du logiciel applicatif, la programmation proprement dite du logiciel et les tests de bas niveau du logiciel (tests unitaires et FAT).

La phase de spécification fonctionnelle traite de la MMRI de façon globale du point de vue de l'exploitant ou du responsable de la maîtrise des risques. La phase de conception détaillée est réalisée par le fournisseur du logiciel et a pour objectif de répondre aux spécifications générales en entrant dans le détail du fonctionnement technique attendu des équipements et du logiciel.

Dans le cadre d'une inspection, il parait peu pertinent d'examiner la conception détaillée. En revanche, il faut s'assurer que l'utilisateur a fourni des exigences suffisamment précises pour obtenir un logiciel fiable, robuste et maintenable.

3.3.1 LA SPÉCIFICATION DÉTAILLÉE DU LOGICIEL

3.3.1.1 SPÉCIFICATION FONCTIONNELLE DU LOGICIEL

La spécification du logiciel permet de faire le lien entre la spécification fonctionnelle de la fonction de sécurité – issue de l'analyse des risques et de l'étude de l'installation – et la programmation du logiciel applicatif. Outre les exigences fonctionnelles elle doit prendre en compte la technologie de l'automate et les bonnes pratiques de traitement des défauts.

La spécification logicielle n'est pas systématiquement formalisée. Bien souvent, le programmeur de l'automate traduit lui-même les spécifications fonctionnelles (spécification générale) en exigences détaillées. Cette démarche est acceptable (mais non recommandée) pour des logiciels de niveau de confiance NC1 si :

- les programmeurs ont une connaissance suffisante de l'installation (code réalisé et maintenu par des équipes sur site) ;
- il existe de bonnes pratiques de programmation identifiant par exemple les traitements des défauts ;
- Le code est écrit de manière organisée et il est commenté. La lecture du code permet d'identifier les éléments de spécification logicielle indiqués ci-après.

La spécification logicielle est conseillée pour des NC1 et nécessaire pour des niveaux de confiance élevés (NC2 ou NC3). Il faut dans ce cas s'assurer de la traçabilité entre les spécifications générales et la spécification logicielle. La spécification logicielle doit comprendre les éléments suivants :

- les types d'entrées sorties et leur plage de valeur ;
- les exigences d'indépendance des entrées sorties ;
- les seuils de déclenchement et la valeur correspondant pour les entrées[6] ;
- l'action de sécurité attendue comprenant les séquences d'arrêts ;
- les conditions d'armement et de réarmement ;

[6] La valeur de mesure définie comme seuil de déclenchement correspond à un niveau de signal pour une entrée analogique ou une valeur pour un signal numérique qui doivent être définis

- le comportement sur défaut ;
- les positions de repli ;
- les types de logiques de votes utilisées ;
- les différents modes de fonctionnement (démarrage, arrêt, maintenance) ;
- les contraintes de temps de réponse.

Il est souhaitable que la spécification du logiciel soit validée par l'exploitant.

3.3.1.2 SPÉCIFICATION DES AUTOTESTS ET DES COMPORTEMENTS SUR DÉFAUT

Des autotests ou tests de diagnostics sont réalisés au démarrage de l'automate et en cours de fonctionnement. Ils visent à détecter les défaillances de l'automate et à atteindre la couverture de diagnostic nécessaire pour le NC ou le SIL requis en fonction de l'architecture matérielle. Ces autotests sont implémentés par le fournisseur de l'automate et sont éventuellement paramétrables par l'utilisateur. Ils sont donc assimilés au logiciel système et ne sont pas évalués avec le logiciel applicatif. L'action déclenchée par les autotests doit toutefois être connue et compatible avec la sécurité de l'installation : ils peuvent par exemple commander un reset de l'automate et une mise en position de repli de toutes les sorties, ce qui peut entrainer une situation dangereuse.

En plus des autotests, les bonnes pratiques de programmation nécessitent de traiter certains défauts dans le logiciel applicatif. On peut par exemple surveiller :

- les valeurs d'entrées hors gamme ;
- les points d'entrée ou de sortie désactivés ou la mise en place de shunt ;
- la perte ou le retard des communications externes relatives à la sécurité ;
- la corruption des communications externes relatives à la sécurité ;
- les divisions par zéro ou autres erreurs logiques.

Les défauts surveillés et les actions sur défauts doivent être spécifiés. La détection d'une panne dangereuse au cours du fonctionnement doit entraîner une action pour :

- mettre en place ou maintenir l'état de sécurité défini pour l'installation ;
- ou si l'automate a une tolérance aux pannes matérielles de un ou plus, réparer la partie en panne pendant la durée moyenne de dépannage (MRT[7]) spécifiée, lorsqu'un fonctionnement continu est autorisé,
- ou si l'automate n'a pas de tolérance aux pannes matérielles et se trouve en mode de faible sollicitation, réparer la partie en panne pendant la durée moyenne de dépannage (MRT) spécifiée.

Le fonctionnement du processus lorsque les systèmes de sécurité sont en mode dégradé exige la mise en place de dispositions supplémentaires de réduction des risques. La durée maximale pendant laquelle l'automate peut demeurer en mode dégradé doit alors être définie. Le MRT doit être inférieur à cette durée.

[7] Mean Repair Time

3.3.1.3 SYNTHÈSE DE LA SPÉCIFICATION LOGICIELLE

Le tableau suivant présente les principaux éléments qui peuvent être vérifiés en spécification logicielle :

Exigences	
Spécification logicielle issue de la spécification système	La spécification du logiciel doit reprendre l'ensemble les exigences système (SRS) applicables et les traduire de manière à pouvoir développer un logiciel applicatif conforme à l'ensemble des exigences système
Description du fonctionnement : Identification des entrées de l'automate, des variables correspondantes et des logiques de vote	La logique peut être traduite sous forme de logigramme, de grafcet, etc.
Seuils de déclenchement	Traduction des seuils de déclenchement capteur (mesure) en seuil de déclenchement automate (valeur du signal)
Identification de l'action de sécurité	Quel est l'état des sorties correspondant à la position de repli ? Quelle est la séquence de déclenchement de cette action ? Quelles sont les temporisations ?
Identification des valeurs d'entrée en défaut ou hors gamme des capteurs et des actionneurs	Les éléments suivants peuvent être signalés et traités comme des défauts : • Valeurs hors échelle de mesure (valeur haute ou basse des entrées) • Signal d'équipement en défaut • Défauts ou retards de communication pour les réseaux de terrain • Défauts d'alimentations des entrées sorties • … Le comportement (mise en repli ou alarme doit être identifié) *Les auto-tests de l'automate sont réalisés par ailleurs et ne font pas partie du logiciel applicatif*
Prise en compte des contraintes matérielles	La conception du logiciel applicatif, prenant en compte l'ensemble des fonctions logicielles de l'automate doit prendre en compte les contraintes matérielles de celui-ci : • Temps de cycle • Charge CPU • Mémoires
Conditions de réarmement	Les conditions de réarmement après activation : il ne doit normalement pas être possible de redémarrer en présence d'un défaut

Inhibition des fonctions de sécurité	Existe-t-il des moyens d'inhibition logiciels, comment sont-ils réalisés, sont-ils limités dans le temps ?
Traçabilité des exigences	La traçabilité entre les spécifications système et les spécifications logicielles doit être vérifiée afin de s'assurer de l'exactitude et de la complétude des spécifications
Plan de tests d'intégration (Plan de FAT)	Les plans de tests d'intégration doivent permettre de tester l'ensemble des conditions d'activation des fonctions de sécurité et des conditions de réarmement et les comportements sur défaut Les plans de tests d'intégration doivent couvrir l'ensemble des combinaisons d'entrées Il n'existe pas toujours de plans de tests détaillés, il convient dans ce cas de s'assurer que des moyens sont mis en œuvre pour réaliser ces tests dans le cadre du développement des logiciels applicatifs. Quoi qu'il en soit, il semble nécessaire de tracer ces tests pour des NC>=2
Exemple de documents	
➢ Spécification logicielle ➢ Logigramme ou algorithme ➢ Plan de test d'intégration	

Tableau 5 : Critères d'évaluation des spécifications détaillées

3.3.2 LA PROGRAMMATION DU LOGICIEL APPLICATIF

La programmation du logiciel applicatif doit suivre des bonnes pratiques afin de garantir la complétude et l'exactitude du logiciel par rapport aux spécifications mais aussi sa fiabilité et sa maintenabilité. Les bonnes pratiques peuvent être définies dans des guides ou référentiels de programmation généraux (à l'installation, à l'entreprise) ou spécifiques à un projet. Si la conception détaillée est sous traitée, l'application de référentiels ou de règles de programmation doit être explicitement mentionné soit par un cahier des charges, soit par l'offre technique du fournisseur de l'application.

Le logiciel applicatif est normalement écrit dans un langage à variabilité limitée (LVL) ou un langage de programme figé FPL. **Lorsqu'un langage de variabilité totale est utilisé, le logiciel doit être évalué conformément à la norme IEC 61508.**

Au minimum, les règles suivantes devraient être vérifiées pour des fonctions de sécurité :

- la programmation du logiciel doit être structurée de manière à obtenir une décomposition modulaire des fonctionnalités. Les fonctions différentes doivent normalement être programmées dans des modules différents. Il

peut cependant y avoir des modules communs à plusieurs fonctions, c'est souvent le cas pour l'écriture des sorties par exemple ;

- l'ensemble des entrées et sorties utilisées pour chacune des fonctions doivent être identifiées. Des règles de nommages doivent être appliquées pour améliorer la lisibilité et la maintenabilité du code ;
- pour les automates de sécurité, des bibliothèques de modules ou de fonctions certifiées sont disponibles (lecture d'une entrée, vote, temporisation, inhibition, écriture d'une sortie , etc.). Le logiciel applicatif doit si possible utiliser ces éléments, si des modules non certifiés ou développés spécifiquement sont utilisés, ils doivent être validés par des tests unitaires ;
- si l'indépendance est requise entre plusieurs fonctions, elles ne doivent pas manipuler les mêmes variables. De même, des fonctions qui ne sont pas de sécurité ne doivent pas modifier des variables utilisées par les fonctions de sécurité.

Dans le cadre d'une inspection, le but ne sera pas de lire le code pour vérifier que ces règles sont appliquées. Cette vérification est un élément du processus de conception à part entière. Le rôle de l'inspecteur sera donc de vérifier, au travers de la documentation du projet, que des bonnes pratiques de programmation sont exigées et que les moyens sont mis en œuvre pour les atteindre.

Pour faciliter la prise en compte des règles de conception, des règles de codage ou des guides de programmation peuvent être définis pour un projet ou au sein d'une entreprise. Les règles de codage comprennent par exemple :

- des règles de nommage unique pour tous les éléments du logiciel (variable, routines, blocs fonctionnels) : tous les éléments doivent être nommés et un nom ne doit pas être utilisé pour plusieurs éléments, etc. ;
- des règles sur la présence de commentaires facilitant la compréhension du code ;
- des règles sur le déroulement d'un cycle automate : les entrées sont lues, la cohérence des entrées est vérifiée, puis les traitements sont réalisées et les sorties sont mises à jour ;
- des règles sur la structure du code : il ne doit pas contenir de code mort et de sous-routine non appelée ;

Un exemple de règle de codage est donné dans le document de référence [8].

Les critères retenus dans les grilles d'évaluation sont les suivants :

Exigences	
Conception modulaire	Le logiciel doit être structuré en modules correspondant à des fonctions simples. Plusieurs architectures sont possibles par exemple : • Plusieurs modules pour chaque fonction de sécurité (entrées / traitement / sorties / gestion des défauts) • Des modules communs à plusieurs fonctions de sécurité (en particulier le module de commande des sorties) • Un module par fonction de sécurité complète Il est recommandé de présenter les modules différents sur des pages de programmation différentes
Langages de programmation	Utilisation de langages à variabilité limitée ou logiciel conforme à la IEC 61508-3
Utilisation de modules certifiés ou de bibliothèques de fonctions certifiées	Il est préférable d'utiliser des bibliothèques certifiées pour les niveaux de confiance élevés Si les modules utilisés ne sont pas certifiés, ils doivent être validés en tests unitaires.
Indépendance des fonctions	Deux fonctions intervenant sur un même scénario ne doivent pas manipuler les mêmes variables
Existence de bonnes pratiques de programmation	Les bonnes pratiques doivent permettre de garantir la fiabilité et la maintenabilité du logiciel Un document de bonnes pratiques (général à l'installation ou spécifique à un projet) peut présenter : • des règles d'utilisation des blocs fonctionnels ; • des règles de nommage des variables ; • des règles sur la structure du programme ; • les règles d'utilisation de la mémoire ; • des règles sur les commentaires.
Exemple de documents	
➢ Règles de programmation ➢ Identification des modules utilisés ➢ Identification des entrées sorties ➢ Logiciel commenté	

Tableau 6 : Critères d'évaluation de la programmation du logiciel applicatif

3.4 THÉME 4 : LES TESTS ET LA VALIDATION

L'objectif de la validation est de s'assurer que le système réalise les services attendus correspondant à ses spécifications (test de conformité de la réalisation aux spécifications), dans un fonctionnement normal ou dégradé. La validation est une étape dans le développement et s'appuie sur des tests.

Différents types de tests sont présentés dans cette partie. Les tests unitaires et les tests d'intégration sont réalisés en usine par le fournisseur du logiciel applicatif. Ils permettent un test approfondi du logiciel. Les tests d'acceptation sur site sont réalisés sur l'installation dans l'environnement de fonctionnement. Ils sont donc plus représentatifs du fonctionnement réel mais tous les essais ne sont pas réalisables sur site.

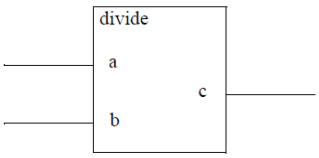
3.4.1 LES TESTS UNITAIRES ET TESTS D'INTÉGRATION (FAT)

Les tests unitaires et les tests d'intégration correspondent à deux types de tests différents qui sont liés au développement de l'automate :

- les tests unitaires sont réalisés lors de la programmation ou du paramétrage d'une fonction logicielle élémentaire (opération algébrique, vérification d'un seuil, vote…) ;
- les tests d'intégration (FAT) sont réalisés sur le programme complet pour tester unitairement chaque fonction ou sous fonction de sécurité.

Les tests unitaires

Les tests unitaires permettent de vérifier unitairement chaque fonction élémentaire. Les fonctions élémentaires sont réalisées par un bloc fonctionnel, ou d'un sous ensemble limité de blocs fonctionnels. Ces tests sont réalisés par le programmeur au cours du développement, ils correspondent à une bonne pratique de programmation. Ils sont nécessaires pour des modules développés spécifiquement pour l'application et ayant un fonctionnement complexe (multiples conditions et paramétrages). Ces tests peuvent être réalisés sur la console de programmation ou sur un émulateur de test. L'objectif est de tester chaque branche des modules.




```
FUNCTION_BLOCK divide
VAR_INPUT
        a: INT;
        b: INT;
END_VAR
VAR_OUTPUT
        c: INT;
END_VAR
        IF b <> 0 THEN
        c := a / b;
        ELSE
        c := 0;
        END_IF;
END_FUNCTION_BLOCK
```

Exemple : *le bloc fonctionnel « divide » ci-dessous est programmé en langage ST (texte structuré). Il réalise la division de a par b et vérifie que b est différent de 0. Différents cas devraient être testés (division normale, b=0, valeurs hors échelle, etc.).*

Les tests d’acceptation en usine (FAT)

Les FAT doivent être réalisés sur une version du logiciel complète et figée et sur un matériel (API) identique à celui installé sur site (y compris le logiciel système). L’automate n’est pas nécessairement relié à ces capteurs actionneurs. Ils sont généralement réalisés par simulation des entrées et lecture des sorties. Ils doivent permettre de tester unitairement les différentes fonctions et chaque branche ou condition de déclenchement de chaque fonction.

Les FAT ne sont pas toujours réalisés. Ils sont recommandés pour des fonctions ayant des logiques complexes ou redondantes qui ne sont pas totalement testables sur site ou pour des NC supérieurs ou égaux à 2.

Un plan de FAT doit couvrir l’ensemble des exigences de spécification détaillée. Il faut en particulier assurer la couverture des différentes conditions de déclenchement, les tests aux limites, les tests de comportement sur défauts.

Pour chaque étape de test les résultats doivent être fournis sous forme de fiches de tests renseignées. Elles doivent identifier :

- la version de la spécification d'essai utilisée;
- les critères d'acceptation des essais d'intégration;
- la version de l’API et du logiciel applicatif faisant l'objet d'essais;
- les outils et équipements utilisés avec des données d'étalonnage;
- les résultats de chaque essai;
- les différences entre les résultats attendus et réels;
- l'analyse réalisée et les décisions prises quant à la poursuite de l'essai ou à l'émission d'une demande de modification (dans les cas où des différences existeraient).

Les non-conformités éventuelles doivent être enregistrées, les actions correctives décrites, avec éventuellement la modification de documentation amont et les fiches de tests suite à correction renseignées.

Si des FAT sont requis, l’utilisateur doit s’assurer qu’ils sont réalisés par le fournisseur.

Les critères d’inspection suivants sont retenus pour les tests unitaires et les FAT dans les grilles d’évaluation :

Exigences	
Réalisation de tests unitaires	Les tests unitaires permettent de valider le bon fonctionnement et le bon paramétrage des blocs fonctionnels élémentaires (vote 1oo2, temporisation, inhibition…) Ils ne sont pas nécessairement tracés mais peuvent faire partie du processus de programmation Ils ne sont pas nécessaires pour les blocs certifiés
Réalisation de tests d'intégration (FAT)	Le plan de FAT doit permettre de tester unitairement chaque fonction avant installation sur site. Les FAT sont réalisées une fois le logiciel applicatif chargé dans l'automate par simulation des entrées et lecture des sorties. Bien souvent les FAT sont réalisées pour des nouveaux projets et pas pour des modifications sur site. (cf. gestion des modifications)
Identification de la version testée	Version logicielle identifiée par un n° de version et/ou un CRC ou un checksum
Identification des équipements utilisés pour le test	Les différents moyens de tests doivent être identifiés. La version de l'automate et des logiciels systèmes de l'automate doivent également être tracés.
Réalisation des tests fonctionnels	Toutes les conditions de déclenchement doivent être testées, y compris pour les logiques de vote Les résultats obtenus doivent être conformes aux résultats attendus et enregistrés
Réalisation des tests sur défaut	Les défauts identifiés en spécifications générales et détaillées doivent être simulés (valeur hors gamme, capteur en défaut, capteur absent, défaut carte d'entrée, etc.) Les résultats obtenus doivent être conformes aux résultats attendus et enregistrés
Exemple de documents	
➢ Fiches de tests renseignées	

Tableau 7 : Critères d'évaluation des tests unitaires et d'intégration

3.4.2 LES TESTS D'ACCEPTATION SUR SITE (SAT)

Après le développement ou la modification d'une fonction de sécurité, une phase de validation par essais est réalisée sur site. Même s'ils ne sont pas identifiés comme SAT (Site Acceptance Test), ces essais sont obligatoires pour les fonctions de sécurité.

Pour ces essais, l'automate est intégré à tout l'environnement nécessaire pour la prise en charge des fonctions de sécurité comprenant tous les capteurs, les actionneurs et les utilités (énergie électrique, source d'alimentation externe, hydraulique, pneumatique…). Il dispose d'une version complète du logiciel applicatif qui a été validée par essais de recette en usine (FAT) et éventuellement tests unitaires.

La SAT doit permettre de vérifier que le développement respecte l'ensemble des spécifications fonctionnelles. Ces essais peuvent également mettre en évidence des défauts de spécification ou de représentation de l'installation. Le plan de SAT permet de tester l'ensemble de la fonction de sécurité, incluant les capteurs, actionneurs, alimentations, etc. Pour couvrir les aspects logiciels, il doit inclure :

- Le test des entrées et des liaisons qui consiste à mesurer que les signaux reçus par l'automate pour les différentes entrées sont conformes aux signaux spécifiés. Cette vérification ne concerne pas le logiciel à proprement dit mais permet de s'assurer que les seuils sont paramétrés aux bons niveaux et de vérifier l'ensemble de la transmission entre la sortie du capteur et l'entrée de l'automate qui peut subir diverses perturbations ;
- le test fonctionnel de l'automate, qui comprend le déclenchement de la fonction de sécurité ;
- les mesures de temps de réponse ;
- les tests sur défaut d'entrée pour des valeurs hors-gamme ou des signaux de défauts, des entrées déconnectées ;
- les fonctions de détections de défauts et les alarmes correspondantes ;
- les déclenchements manuels ;
- le réarmement après déclenchement ;
- la signalisation et l'affichage appropriés des alarmes et des déclenchements.

Il n'est pas possible en SAT de couvrir l'ensemble des conditions de déclenchement, ce qui serait trop couteux en temps à ce stade (par exemple une architecture 2oo3 demanderait de provoquer 3 déclenchements et réarmements de la fonction de sécurité). Il est préférable de réaliser ce type d'essais lors des FAT.

Des fiches d'essais doivent être renseignées et permettre d'identifier :

- la version du plan d'essais qui a été utilisée ;
- les versions logicielles et matérielles testées ;
- la fonction instrumentée de sécurité en essais (ou en analyse), avec la référence spécifique ;

- les outils et les équipements utilisés, avec les données d'étalonnage ;
- les résultats de chaque essai ;
- les critères d'acceptation de chaque essai;
- toute divergence entre les résultats attendus et les résultats réels.

A l'issue de ces différentes étapes de validations des SIF, les fiches de tests doivent être archivées par l'exploitant dans un dossier de suivi de l'automate.

En cas de résultat non conforme, une analyse doit être réalisée, les corrections nécessaires apportées et les tests nécessaires réalisés.

A la fin de la validation, lorsqu'il n'y a plus d'écarts ou qu'ils sont tous justifiés, le logiciel chargé dans l'automate doit être considéré comme validé et aucune modification ne doit être entreprise sans le respect d'une procédure spécifique.

Pour un SIL 3 ou un NC 3 les responsables de la validation doivent appartenir à un service ou à une organisation indépendante des équipes de développement.

Les exigences minimales pour la réalisation des SAT sont données dans le tableau ci –dessous :

Exigences	
Réalisation de SAT (Tests sur site des fonctions de sécurité)	Les SAT sont réalisées sur site, l'automate étant connecté à ses capteurs et actionneurs. Le but est de réaliser l'ensemble des essais non destructifs de la fonction de sécurité. Les SAT sont en général réalisées pour la MMRI dans son ensemble (du capteur à l'actionneur). Il faut veiller à ce que les SAT permettent de vérifier les différents points de la spécification fonctionnelle du logiciel. L'objectif est de vérifier que la fonction de sécurité est conforme à la spécification générale.
Identification de la version testée	Version logicielle identifiée par un n° de version et/ou un CRC ou un checksum
Réalisation de tests fonctionnels	Test des conditions d'activation : logiques de vote, réaction sur les différents seuils d'alarme et de déclenchement. Si possible (si ça ne génère pas de risque), ces tests sont réalisés en générant le phénomène physique au niveau du capteur.
Mesure des temps de réponse	Temps de réponse mesurés avec une précision suffisante pour le temps de réponse attendu
Réalisation de tests sur défaut	Il s'agit de tests non destructifs : perte d'un capteur, signal hors échelle, pertes d'alimentation…
Exemple de documents	
➢ Fiches de tests renseignées	

Tableau 8 : Critères d'évaluation des SAT

3.5 THÈME 5 : LE SUIVI DU LOGICIEL EN EXPLOITATION

La gestion des versions et des modifications ainsi que les tests périodiques doivent permettre d'assurer que les versions en exploitation correspondent aux versions validées, de tracer les différentes évolutions du logiciel et de maintenir ou d'améliorer la fiabilité de la fonction au cours des différentes modifications.

3.5.1 LA GESTION DES VERSIONS

Pour chaque module logiciel, il doit être possible d'obtenir les informations suivantes :

- identification si possible au moyen d'une signature (CRC) et date de mise en service ;
- statut (en développement, validé, abandonné) ;
- version en cours de validité ;
- documentation relative au développement depuis le dossier de spécification technique jusqu'aux résultats de tests ;
- versions des outils logiciels utilisés pour le développement ;
- identification des personnes intervenues sur le logiciel ;
- configuration matérielle et logicielle (version d'OS) pour laquelle la validation a été effectuée ;
- enregistrement des versions avant modification : les versions successives doivent être archivées, il doit être possible de revenir au minimum à la version validée précédente.

Exigences	
Identification des versions installées	Identification par un n° de version et une date des versions en cours de validité Identification du n° de version de chaque fonction ou de chaque page de programmation Identification par un CRC ou une signature
Historique des versions	L'ensemble des versions doit être identifiable par une date et un statut (en cours de validité, en développement, abandonné)
Information sur les versions	Les modifications réalisées entre les différentes versions doivent être expliquées Les outils logiciels utilisés pour le développement, document de conception et de validation, personnes intervenues pour la modification et configuration matérielle pour laquelle la version est valable doivent être identifiés
Archivage des versions	L'archivage des versions doit permettre de restaurer au minimum la version antérieure et la version en cours de validité Elles doivent être archivées dans deux lieux différents (il est recommandé de disposer d'un exemplaire sur site)
Exemple de documents	
➢ Procédure des gestions de versions ➢ Archivage des différentes versions	

Tableau 9 : Critères d'évaluation de la gestion de versions

3.5.2 LA GESTION DES MODIFICATIONS

La procédure de gestion des modifications doit empêcher toute modification non autorisée et non maitrisée. Pour cela il faut s'assurer des éléments suivants :

- toute modification doit être enregistrée, une demande de modification doit en préciser les raisons et les impacts possibles sur la sécurité, le reste du système et la documentation ;
- avant de modifier le logiciel, le programme courant doit être archivé ;
- après avoir réinstallé le logiciel modifié, un essai de réévaluation doit être effectué, garantissant que la sécurité n'a pas été dégradée ;
- les modifications du logiciel doivent être soumises au même cycle de vie que le développement du logiciel initial, y compris pour les activités concernant la documentation ;
- seules les personnes autorisées doivent avoir la possibilité de réaliser les modifications, si possible, les logiciels doivent être protégés par un système de verrouillage ou un mot de passe.

Exigences	
Procédure de gestion des modifications du logiciel	Toute modification doit faire l'objet d'une demande de modification, identifiant les raisons et analysant l'impact sur la sécurité L'analyse d'impact doit évaluer l'impact des modifications d'une fonction sur les autres fonctions de sécurité de l'automate Les personnes responsables d'autoriser et de valider la modification doivent être identifiées
Archivage de la version courante du logiciel avant modification	L'archivage doit être réalisé et récupérable
Cycle de vie des modifications	En fonction des résultats de l'analyse d'impact, les modifications doivent suivre totalement ou en partie le processus de développement et validation que le logiciel initial
Protection contre les modifications	Seules les personnes autorisées doivent avoir la possibilité de réaliser des modifications Les logiciels critiques doivent être protégés par un système de verrouillage ou mot de passe
Validation de la modification	Des essais de réévaluation doivent permettre de s'assurer que la sécurité n'a pas été dégradée. Ils doivent couvrir les phases de tests pertinentes suivant l'importance de la modification (FAT, SAT, temps de réponse, acquisition d'une mesure). Les plans de tests doivent être cohérents avec les plans de FAT et SAT précédents si des FAT ou SAT sont réalisées en modification.
Exemple de documents	
➢ Procédure de gestion des modifications ➢ Analyses d'impact ➢ Documents de validation de modifications	

Tableau 10 : Critères d'évaluation de la gestion des modifications

3.5.3 LES TESTS PÉRIODIQUES

Les tests périodiques doivent permettre de s'assurer que la MMR conserve ses performances dans le temps. Ils concernent principalement les défaillances ou vieillissement du matériel. Des modifications du matériel (par exemple remplacement d'un capteur par une nouvelle référence ayant des valeurs de signaux de sorties différents) peuvent avoir un impact sur le logiciel.

Pour le logiciel, vérifier que les versions en exploitation correspondent aux versions validées, permet de s'assurer que la gestion des modifications est maitrisée et qu'il n'y a pas de dégradations du niveau de sécurité du logiciel. Cela peut être vérifié par un relevé de versions. La méthode la plus efficace consiste à utiliser la signature (de type CRC) du logiciel.

En test périodique, la réalisation des tests fonctionnels et sur défaut de la fonction de sécurité permet de vérifier les aspects logiciels.

Exigences	
Réalisation d'un relevé des versions	Vérification que les versions implémentées correspondent aux versions validées
Réalisation de tests fonctionnels et sur défaut	Ces tests correspondent aux tests réalisés en SAT. Ils permettent de vérifier la non régression du logiciel
Exemple de documents	
➢ Compte rendu de relevé de versions ➢ Plans de tests périodiques ➢ Fiche de tests périodiques renseignée	

4. PRINCIPE ET DÉROULEMENT DE L'INSPECTION

4.1 UNE INSPECTION BASÉE SUR LES FONCTIONS ET LE CYCLE DE VIE

L'évaluation d'un logiciel critique se fait au travers de l'évaluation de son cycle de vie. On cherche à s'assurer que chaque étape du cycle de vie a été réalisée en appliquant des bonnes pratiques et que les exigences relatives à la MMRI sont suivies tout au long du cycle de vie. L'objectif est d'éviter que des erreurs ne soient introduites dans le développement du logiciel ou en cours d'exploitation.

Pour cela, la fiche d'inspection proposée en annexe 4 suit les différentes étapes de ce cycle et les processus supports mis en œuvre.

Cette fiche présente des critères détaillés et les justifications à rechercher. L'évaluation de certains points peut être plus ou moins approfondie suivant le niveau de confiance recherché. Les grilles précisent si les critères sont pertinents pour des NC1, NC2, ou NC3.

Dans les faits, les différentes étapes sont rarement documentées de manière formalisée. Il faudra néanmoins s'assurer que les exigences techniques ou organisationnelles sont satisfaites pour les étapes de spécification, réalisation, validation et gestion des modifications.

Certaines étapes pouvant être externalisées, l'accès à l'ensemble de la documentation n'est pas toujours possible. Il convient néanmoins de s'assurer que l'exploitant maîtrise la sous-traitance.

Le cycle de vie sera inspecté au travers de son application sur une ou plusieurs fonctions représentatives. Le but n'est pas de valider l'ensemble des logiciels mais de vérifier par échantillonnage que les moyens nécessaires sont mis en œuvre par l'exploitant.

L'inspection ne peut pas analyser de manière détaillée chacun des points présentés au chapitre précédent, qui relèvent de bonnes pratiques de programmation. Elle doit permettre de s'assurer d'une maîtrise globale des logiciels compatible avec les exigences applicables aux MMRI.

4.2 PRÉPARATION DE L'INSPECTION

Dans la phase préparatoire, on cherchera à identifier les MMRI mises en œuvre sur le site, leurs technologies et les personnes intervenant dans leur cycle de développement.

Pour cela, différents documents peuvent être demandés :

- une liste des MMRI du site, avec éventuellement des fiches d'évaluation ;
- le processus de développement des logiciels applicatifs ou de sous-traitance de ces développements.

Identification des MMRI

L'identification des MMRI peut se faire sur les nœuds papillons ou dans un document descriptif des barrières de sécurité. On cherchera à identifier les

automates réalisant les traitements de ces différentes fonctions. Il peut s'agir d'automates de sécurité ou d'automates de contrôle-commande. Différents cas de figures sont possibles :

1. Automate de sécurité ne réalisant pas plus d'une fonction par scénario.
2. Automate réalisant plusieurs fonctions de sécurité intervenant sur un même scénario.
3. Automate de conduite réalisant une fonction de sécurité.
4. Automate de conduite réalisant plusieurs fonctions sur un même scénario : ce cas n'est normalement pas autorisé par la doctrine MMRI.

Les cas 2 et 3 sont plus intéressants du point de vue de l'évaluation car ils permettent de vérifier le respect de règles d'indépendance au niveau du logiciel.

Si un seul automate de sécurité est utilisé pour plusieurs fonctions, sur le même scénario ou non, on basera l'évaluation sur la fonction la plus pertinente en fonction du contexte de l'analyse. Il pourra s'agir de la fonction la plus complexe, ou de celle de NC le plus élevé par exemple.

L'objectif de cette étape est de définir une ou deux MMRI représentatives qui serviront de support à l'évaluation.

Données générales sur l'organisation

On cherche à identifier les personnes et les processus à prendre en compte pour la maîtrise des logiciels applicatifs. L'organisation mise en place sera précisée :

- Quelle est la démarche générale pour la validation, la maintenance, les modifications des logiciels applicatifs ? (cycle de vie)
- Quel est le processus qualité appliqué pour le développement et la validation des logiciels applicatifs ?
- Quels services / entreprises sont impliquées dans le cycle de vie ?
- Quels référentiels de sûreté de fonctionnement ou de sécurité fonctionnelle sont appliqués ?
- Quelles règles et bonnes pratiques de programmation sont définies ?

Les informations sur le contexte technique ne visent pas à vérifier le respect d'exigences mais à faire un état des lieux de l'existant sur l'installation et à identifier les MMRI qui serviront de support à l'inspection.

L'exploitant doit être en mesure de lister les MMRI de son installation, la fonction de sécurité associée et le Facteur de Réduction de Risque ou Niveau de confiance requis.

Les informations sur les technologies utilisées permettent également de cadrer l'inspection. Les MMRI peuvent être implémentées sur des automates de sécurité, des centrales feu et gaz ou des automates de contrôle commande. Un automate de sécurité peut exécuter plusieurs fonctions de sécurité qui peuvent intervenir sur le même scénario d'accident ou sur des scénarios différents.

4.3 ORGANISATION DE L'INSPECTION

L'inspection sera organisée de manière à obtenir les informations sur les différents processus mis en œuvre et les exigences techniques requises.

Lors de l'inspection, des personnes compétentes pour répondre aux questions sur les différentes phases du cycle de vie devront être présentes, c'est-à-dire des personnes compétentes pour :

- la spécification générale des fonctions de sécurité ;
- les exigences spécifiées pour le développement des logiciels applicatifs ;
- la validation des logiciels ;
- la maintenance, la gestion des versions et des modifications.

4.4 CONTENU DE L'INSPECTION

L'inspection suit le cycle de vie de manière chronologique. Par rapport aux grilles détaillées à destination des exploitants, la fiche d'inspection est simplifiée principalement pour les parties relevant de la responsabilité du fournisseur : on cherche à s'assurer que l'utilisateur, qui est responsable de la validation globale de la MMRI, fournit les données nécessaires et spécifie les exigences suffisantes pour s'assurer de la qualité du logiciel fourni.

L'inspection est structurée autour de 5 questions générales pour lesquelles les grilles de l'annexe 4 fournissent des éléments d'appréciation précis :

- Q1 : Quel est le lien entre la spécification fonctionnelle détaillée de la MMRI et le développement du logiciel applicatif ?
- Q2 : Quelles bonnes pratiques de programmation sont définies ?
- Q3 : Comment le plan de validation permet-il de vérifier l'ensemble des exigences ?
- Q4 : Comment s'assure-t-on que les performances du logiciel sont maintenues dans le temps ?
- Q5 : Comment les processus relatifs au logiciel permettent-ils d'atteindre et maintenir un niveau de sécurité ?

L'inspection simplifiée est prévue pour être réalisée en 2 à 4 heures selon la profondeur de l'évaluation et la facilité à trouver les informations.

4.5 CONCLUSIONS DE L'INSPECTION

L'inspection doit vérifier la maîtrise globale du logiciel par l'industriel au travers des différentes questions. Les premières actions minimales à mettre en place par un industriel qui n'aurait pas initié une démarche de maîtrise des logiciels applicatifs seraient :

- description fonctionnelle des MMRi ;
- tests d'acceptation et périodiques des MMRi conformément à la description fonctionnelle ;

- relevé des versions logicielles en exploitation et mise en place d'une gestion des modifications.

Dans un second temps, l'ensemble des points de la démarche présentée devraient être appliqués, en particulier pour les nouveaux développements.

5. CONCLUSION

5.1 SYNTHÈSES SUR LES BONNES PRATIQUES DE DÉVELOPPEMENT DES LOGICIELS APPLICATIFS

Les mesures de maitrise des risques font de plus en plus appel à des systèmes instrumentés et à des automates programmables. Il appartient à l'exploitant d'un site d'avoir la maitrise de ses systèmes ; cela passe par la maitrise des logiciels applicatifs.

Les logiciels, et les boucles instrumentées de manière générale sont souvent fournis clé en main par des sous-traitants. Il convient néanmoins de s'assurer qu'ils sont conformes aux fonctions et au niveau de performance attendus. Pour cela, on s'appuie sur un cycle de vie qui permet de suivre les exigences au travers de différentes étapes. L'exploitant devra au minimum spécifier la fonction, valider le système fourni et s'assurer de son maintien dans le temps. De plus, il est souhaitable de définir des exigences de développement et de test à destination des fournisseurs.

Les grilles présentées en annexe 3 permettent de suivre ces exigences.

5.2 SYNTHÈSE SUR L'INSPECTION

L'objectif de l'inspection n'est pas de valider en détail le logiciel applicatif et son niveau de confiance mais de vérifier sur des fonctions représentatives de l'installation que les moyens nécessaires au développement, à la validation et au maintien d'un logiciel fiable sont mis en œuvre.

L'inspection sera donc basée sur les documents de conception, de développement et de validation permettant de suivre l'application des exigences sur une fonction de sécurité tout au long de son cycle de vie.

On cherchera donc à obtenir au minimum les éléments permettant de répondre aux 5 questions ci-dessous :

1. Comment est réalisée la spécification fonctionnelle du logiciel ?
    - les spécifications du logiciel doivent être issues des spécifications fonctionnelles ;
2. Quelles bonnes pratiques de programmation sont- appliquées ?
    - des bonnes pratiques de programmation doivent être appliquées ;
    - les équipes de développement doivent être compétentes pour la technologie de l'automate et la sécurité.
3. Comment le plan de validation permet-il de vérifier l'ensemble des exigences ?
    - des procédures de validation et d'acceptation doivent être mises en place et intégrer les organisations concernées ;
    - que le développement soit interne ou externe, une personne différente du développeur doit participer à la validation ;

- l’exploitant est responsable de l’acceptation finale du logiciel.

4. Comment s’assure-t-on que les performances du logiciel sont maintenues dans le temps ?
    - une gestion des modifications et des versions doit être mise en place ;
5. Quel est le processus relatif au logiciel de sécurité ?
    - une organisation doit être mise en place pour la spécification et l’acceptation des MMRI dans leur ensemble. Elle doit prendre en compte les aspects logiciels ;
    - Ce processus est-il appliqué pour tous les logiciels applicatifs du site ? Depuis quand ? Quelles sont les fonctions de sécurité réalisées par des logiciels développés avant cette date ?

Les grilles d’évaluations détaillée et simplifiée fournissent des éléments précis permettant de justifier la réponse à ces 5 questions en s’appuyant sur le cycle de vie du logiciel applicatif et les responsabilités des différents intervenants.

Les conclusions de l’inspection doivent permettre de hiérarchiser les actions à mettre en œuvre selon le niveau de maturité de l’industriel sur le sujet des logiciels applicatifs (cf. 4.5).

6. DOCUMENTS DE RÉFÉRENCE

[1] Accidentologie des automatismes industriels partie 2/3 : La fonction traitement, BARPI, 2014

[2] Norme IEC 61511 :2011 - Sécurité fonctionnelle - Systèmes instrumentés de sécurité pour le domaine de la production par processus

[3] Normes IEC 61131-3 :2013 - Automates programmables - Partie 3 : Langages de programmation

[4] Norme IEC 61131-6 - Automates programmables - Partie 6 : sécurité fonctionnelle

[5] Méthode Ω10 - Evaluation des Barrières Techniques de Sécurité, INERIS, 2018

[6] Note de doctrine relative aux Mesures de Maîtrise des Risques instrumentées (MMRI), GT MMRI, 2013

[7] Guide DT 93 : Guide méthodologique pour la gestion et la maîtrise du vieillissement des Mesures de Maîtrise des Risques Instrumentées (MMRI)

[8] PLC programs development guidelines – Itris Automation Square, 2014

7. LISTE DES ANNEXES

Repère	Désignation	Nombre de pages
Annexe 1	Glossaire	3
Annexe 2	Langages de programmation	2
Annexe 3	Fiche détaillée	18
Annexe 4	Fiche d'inspection	10
Annexe 5	Liste de documents types	2
Annexe 6	Proposition de plan d'inspection	1

ANNEXE 1 GLOSSAIRE

API *:* Automate Programmable Industriel

Checksum : Somme de contrôle

CPU : Control Processing Unit (processeur)

CRC : Cyclic Redundancy Check (contrôle de redondances cyclique)

BPCS : Basic Process Control System

Système de contrôle commande du processus

FAT : essai d'acceptation en usine

Activité visant à démontrer que le système du fournisseur et les autres systèmes fournis sont conformes à la spécification

SIF : Safety Instrumented Function (Fonction instrumentée de sécurité)
Fonction de sécurité réalisée par un système instrumenté de sécurité

SIS : Système Instrumenté de Sécurité
Système Instrumenté utilisé pour réaliser une ou plusieurs SIF

Fonction de sécurité : Fonction à réaliser par un système SIS, par un système relatif à la sécurité basé sur une autre technologie, ou par des dispositifs externes de réduction de risque, prévue pour assurer ou maintenir un état de sécurité au processus, par rapport à un événement dangereux spécifique.

IHM : Interface Homme Machine

Ensemble de dispositifs matériels et logiciels permettant à un utilisateur de communiquer avec un système informatique.

Logiciel : Création intellectuelle comprenant les programmes, les procédures, les données et les règles, ainsi que toute documentation associée se référant au fonctionnement d'un système de traitement de données.

Logiciel applicatif : Logiciel spécifique à l'application de l'utilisateur. En général, il contient les séquences logiques, seuils et expressions qui contrôlent les entrées, sorties, calculs et décisions nécessaires pour atteindre les exigences fonctionnelles du SIS.

MMR : Mesure de Maîtrise des Risques

Les MMR sont définies dans le cadre des études de dangers dans un objectif de prévention des accidents majeurs.

MMRI : Mesure de Maîtrise des Risques Instrumentée

Une MMRI est une MMR faisant appel à de l'instrumentation de sécurité et constituée d'un ensemble d'éléments techniques et/ou organisationnels nécessaires et suffisants pour assurer une fonction de sécurité. Elle est constituée par une chaîne de traitement comprenant une prise d'information (capteur, détecteur…), un système de traitement (automate, calculateur, relais…) et une action (actionneur avec ou sans intervention d'un opérateur).

MMRIC : Mesure de Maîtrise des Risques Instrumentée de Conduite

Une MMRIC est une MMRI intégrée au système de conduite.

MMRIS : Mesure de Maîtrise des Risques Instrumentée de Sécurité

Une MMRIS est une MMRI intégrée au système de sécurité.

MRT : Mean Repair Time

Temps moyen de réparation

NC : Niveau de confiance

Classe de probabilité pour qu'une MMRI, dans son environnement d'utilisation, n'assure pas la fonction de sécurité pour laquelle elle a été choisie. Cette classe de probabilité est déterminée pour une efficacité et un temps de réponse donnés. Ce niveau de confiance est issu des SIL (Safety Integrated Level) définis dans les normes NF EN 61508 et NF EN 61511.

SAT : essai d'acceptation sur site

Activité visant à démontrer que l'installation des différents systèmes du fournisseur est conforme aux spécifications applicables et aux instructions d'installation

SIT : essai d'intégration sur site

Activité visant à démontrer que la fusion des différents systèmes en un seul système global est terminée et que tous les composants fonctionnent ensemble comme prévu

SRS : Spécification des exigences de sécurité (Safety Requirement Specification)

SIL : Safety Integrity Level

Niveau discret (parmi quatre possibles) permettant de spécifier les exigences concernant l'intégrité de sécurité des fonctions instrumentées de sécurité, à allouer aux systèmes instrumentés de sécurité. Le niveau d'intégrité de sécurité 4 possède le plus haut degré d'intégrité ; le niveau 1 possède le plus bas.

SIS : Système Instrumenté de sécurité

Système instrumenté utilisé pour mettre en œuvre une ou plusieurs fonctions instrumentées de sécurité. Un SIS se compose de n'importe quelle combinaison de capteur(s), d'unité(s) logique(s) et d'élément(s) terminal(aux).

ANNEXE 2 LES LANGAGES DE PROGRAMMATION

Les automates sont programmés par l'intermédiaire d'outils spécifiques au fournisseur et/ou à la gamme. Différents langages de programmation peuvent être proposés. La norme 61131-3 définit 4 langages de programmation pouvant être utilisés par les automates qui sont de deux types : textuels ou graphiques.

Les langages textuels définis sont IL (Instruction List) et ST (Structured Text) :

- Le langage IL est proche de l'assembleur. Il utilise les ressources matérielles de bas niveau du processeur. Ce langage est aujourd'hui dépassé mais peut être utilisé sur des automates anciens. Siemens a développé un langage dérivé d'IL appelé STL (Statement List).
- Le langage ST est un langage de haut niveau dont la structure est proche de celle des langages Ada ou Pascal. Il utilise des boucles d'itération (REPEAT-UNTIL ; WHILE-DO), des conditions (IF-THEN-ELSE ; CASE), et des fonctions.

Les langages graphiques définis sont LD (Ladder Diagramm) et FBD (Function Bloc Diagramm)

- Le ladder diagramm représente les fonctions logiques sous forme de schémas électriques. Il ressemble aux schémas utilisés pour décrire les logiques à relayage. On parle également de langage à contact ou schéma à contact. Cette représentation, populaire auprès des automaticiens, tend à être de moins en moins utilisée dans l'industrie.
- Le FBD est un langage constitué de blocs fonctionnels qui décrivent une fonction entre à gauche des entrées et à droite des sorties. Les blocs peuvent avoir des fonctions plus ou moins complexes (fonctions logiques, fonction mathématiques, temporisation). Les sorties d'un bloc peuvent être reliées à des entrées d'autres blocs.

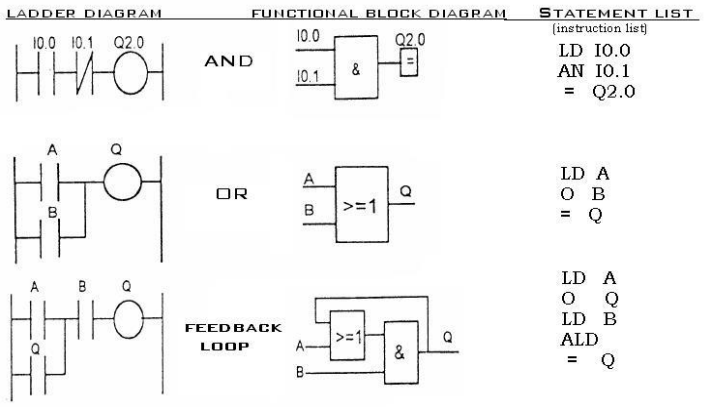


Figure 1 : Exemple de représentation en ladder diagramm, FBD et IL

L'ensemble de ces langages permettent de commander les sorties en fonction de l'état des entrées, de respecter les exigences de séquencement et de temps de réponse.

Les langages graphiques définis par la norme 61131-3 (LD, FBD, SFC) sont des langages à variabilité limitée (LVL) au sens de la norme IEC 61511. Ils sont donc recommandés pour le développement des fonctions de sécurité.

ANNEXE 3 FICHE DÉTAILLÉE

Fiche d'évaluation détaillée des logiciels applicatifs

<table>
<tr><td colspan="2">Site ou système évalué :</td><td colspan="2">Date :</td></tr>
<tr><td rowspan="3">Thème de la visite :</td><td rowspan="3">MMRi – Développement, validation et maitrise des logiciels applicatifs</td><td>Type de visite d'inspection : Approfondie</td><td rowspan="2">Responsable de l'évaluation :</td></tr>
<tr><td>Secteur industriel :</td></tr>
<tr><td>Type d'installation :</td><td rowspan="2">Autres participants :</td></tr>
<tr><td colspan="2">Référentiel :</td></tr>
</table>

Grilles d'évaluation « LOGICIELS APPLICATIFS DES MMRi »

Cette fiche comprend est destiné à mettre en œuvre et évaluer de façon détaillée un processus de maîtrise des logiciels applicatifs adapté aux barrières techniques de sécurité. Elle est avant tout destinée à un exploitant qui souhaite spécifier une ou plusieurs fonctions de sécurité et s'assurer que le processus de développement et de validation permet de lui accorder une confiance justifiée.

Cette fiche comprend 4 grilles d'inspection autour des thèmes suivants :

- ✓ Thème n°1 : Description du contexte et de l'organisation (responsabilité de l'exploitant)
- ✓ Thème n°2 : Spécification et validation globales (données d'entrée de l'exploitant pour un projet particulier)
- ✓ Thème n°3 : Conception et vérification détaillées du logiciel applicatif (responsabilité du fournisseur)
- ✓ Thème n°4 : Suivi du logiciel en exploitation (responsabilité de l'exploitant)

- ➢ Les grilles ne proposent pas de liste type des documents à demander, la gestion documentaire étant variable d'une entreprise à une autre. Elles sont structurées selon le cycle de vie du logiciel et les informations à fournir à ses différentes étapes. Il faut garder à l'esprit qu'on cherche à évaluer les performances propres au logiciel applicatif dans le cadre plus large des performances globales d'une MMRI.
- ➢ Les grilles indiquent si les exigences sont applicables pour des fonctions NC1, NC2 ou NC3. Cette indication est faite pour ne pas être excessivement pénalisante. Cependant, un processus d'ingénierie des logiciels devrait être mis en place quel que soit le niveau de confiance visé et l'ensemble des exigences devrait être respecté. Si des FIS de niveaux de confiance différents sont traitées par le même automate, l'ensemble du logiciel applicatif doit être réalisé en respectant les exigences du NC le plus élevé.
- ➢ Les grilles 1, 2 et 4 présentent des exigences de haut niveau applicables par l'utilisateur du logiciel (l'exploitant). La grille 3 présente des exigences sur le développement du logiciel applicables par le développeur (fournisseur) du logiciel.

Fiche d'évaluation détaillée des logiciels applicatifs

Informations nécessaires à la réalisation de l'évaluation :

- ✓ Identification des MMRi de l'installation
- ✓ Les spécifications des fonctions de sécurité instrumentées ;
- ✓ Si les développements sont externalisés le cahier des charges (ou documents équivalents servant à définir la commande pour le développement des logiciels applicatifs) ;
- ✓ Les Règles ou bonnes pratiques de développement des logiciels applicatifs appliquées ;
- ✓ Les plans et résultat de tests de validation d'une fonction de sécurité instrumentée ;
- ✓ Les procédures de gestion des versions et des modifications.

Lors de l'inspection, les personnes compétentes pour les différentes phases du cycle de vie du logiciel devront être présentes :

- ✓ Spécifications fonctionnelles ;
- ✓ Développement détaillé ;
- ✓ Validation des logiciels applicatifs ;
- ✓ Maintenance et modification des automates et logiciels.

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 1-CONTEXTE TECHNIQUE ET ORGANISATION						
n°	Questions	Éléments de discussion	Applicable			Explications de l'exploitant et commentaires de l'inspection
			NC 1	NC 2	NC 3	
CONTEXTE TECHNIQUE						
1	Présentation des MMRi de l'installation	Identification des MMRI de l'installation. Identification de leur Niveau de confiance. Identification des types d'équipement utilisés (automates de sécurité, automates de contrôle commande, centrales feu et gaz). *On cherchera à avoir une vision de l'existant sur l'installation et à sélectionner une ou deux fonctions pertinentes comme support à l'inspection.*	X	X	X	
2	Y-a-t-il des référentiels de développement appliqués ?	Des normes appliquées pour le développement ou la certification des logiciels applicatifs font elles apparaitre des phases de spécification, conception, validation. *Pour des logiciels de sécurité des installations classées, les normes pertinentes sont principalement la norme CEI 61511 et la norme CEI 61508.* *Le développement et la validation des logiciels peuvent également être intégrés au plan d'assurance qualité (ISO 9001) de l'entité inspectée.* *L'intérêt du référentiel est de faire apparaitre les différentes étapes du cycle de vie et des exigences associées.*		X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 1-CONTEXTE TECHNIQUE ET ORGANISATION						
n°	Questions	Éléments de discussion	Applicable			Explications de l'exploitant et commentaires de l'inspection
			NC 1	NC 2	NC 3	
3	Antériorité du logiciel	Antériorité et vie du logiciel (ancien projet peu documenté, projet récent…). *Pour des logiciels très anciens, l'exploitant peut avoir perdu la maitrise du développement et la connaissance approfondie du développement. En revanche un retour d'expérience, même informel peut exister.*	X	X	X	
4	Retour d'expérience sur les MMRI	• Retour d'expérience sur la sollicitation des MMRI. • Y-a-t-il eu des déclenchements intempestifs des MMRI ? • Dans quelles circonstances ? • Est-ce tracé ?	X	X	X	
CYCLE DE VIE ET ORGANISATION						
5	Cycle de vie appliqué	• Description du cycle de vie et des différentes étapes de spécification, de développement, de validation et d'exploitation du logiciel.	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 1-CONTEXTE TECHNIQUE ET ORGANISATION						
n°	Questions	Éléments de discussion	Applicable			Explications de l'exploitant et commentaires de l'inspection
			NC 1	NC 2	NC 3	
6	Responsabilités et compétences pour le développement des logiciels des MMRI	• Entités ou personnes responsables de la maîtrise de différentes phases du cycle de vie (spécification, réalisation, validation, maintenance, modification) ; • Personnes et entités impliquées dans la réalisation des différentes phases (sous-traitance, compétence interne) ; • Compétence sur les technologies mises en œuvre (équipements et outils de programmation utilisés).		X	X	
7	Indépendance entre les responsables des différentes phases	Indépendance entre la personne réalisant le logiciel applicatif et la personne réalisant des tests d'intégration (FAT).	X	X	X	
		Indépendance entre la personne réalisant le logiciel applicatif et la personne réalisant des tests de validation (SAT).	X	X	X	
		Le responsable de l'acceptation globale doit être l'utilisateur.	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 2-SPÉCIFICATIONS FONCTIONELLES ET VALIDATION GLOBALE						
n°	Questions	Éléments de discussion	Applicable			Explications de l'exploitant et commentaires de l'inspection
			NC 1	NC 2	NC 3	
SPÉCIFICATIONS FONCTIONNELLES						
1	Existence d'une spécification système formalisée	Exemple SRS ou cahier des charges.		X	X	
2	Identification de la fonction de sécurité et du scénario envisagé	L'objectif est de partir d'une spécification de très haut niveau de la fonction dans son ensemble. On commence par identifier les MMRi intervenant sur un scénario et leur NC. *Rq : Si plusieurs fonctions d'un même scénario sont traitées par le même automate les règles d'indépendance devront être vérifiées.*	X	X	X	
3	Identification des entrées et sorties Tolérance aux défaillances nécessaires pour atteindre le NC souhaité	Identification des capteurs et des points de mesure, identification des types d'entrée (analogique, numérique, TOR). Identification des actionneurs et de leur mode d'activation (à manque, à émission).	X	X	X	
3bis	Indépendance des entrées sorties Pour un BPCS ou un automate réalisant plusieurs fonctions de sécurité sur un même scénario	Les fonctions indépendantes (conduite vs sécurité, sécurité vs sécurité) doivent utiliser des cartes d'ES différentes. L'indépendance est rarement réalisée pour les sorties : plusieurs fonctions indépendantes peuvent partager leurs sorties. 2 fonctions indépendantes ne doivent pas avoir des positions de sorties antagonistes. Les fonctions de sécurité doivent être prioritaires sur les fonctions de conduite pour la commande des actionneurs pouvant être partagés.	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 2-SPÉCIFICATIONS FONCTIONELLES ET VALIDATION GLOBALE						
n°	Questions	Éléments de discussion	Applicable			Explications de l'exploitant et commentaires de l'inspection
			NC 1	NC 2	NC 3	
4	Conditions et seuils de déclenchement de l'action de sécurité	Quelle est la logique de vote entre les capteurs ? Quels sont les seuils de déclenchement ? Y-a-t-il des temporisations ?	X	X	X	
5	Définition de l'action de sécurité	Quels actionneurs sont sollicités ? Séquence de commande des actionneurs, temporisations éventuelles. Type de commande des actionneurs (TOR à manque ou à émission, numérique). Identification des alarmes (buzzer, voyants, etc.).	X	X	X	
6	Comportement sur défauts	Les signaux d'alarme des capteurs et les valeurs hors échelle doivent être traités (alarme pour mise en place de moyens compensatoires ? déclenchement ? etc.), pour des signaux d'entrée redondants il peut y avoir des tests de cohérence. Quels comportements sont attendus sur ces défauts ? de même, des moyens de diagnostics peuvent être intégrés aux actionneurs, comment sont-ils traités (ex pour les vannes : positionneurs, fin de course ou partial stroke test).	X	X	X	
7	Exigences de temps de réponse	Temps de réponse de la MMR globale et temps de réponse attendu pour l'automate.	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 2-SPÉCIFICATIONS FONCTIONELLES ET VALIDATION GLOBALE						
n°	Questions	Éléments de discussion	Applicable			Explications de l'exploitant et commentaires de l'inspection
			NC 1	NC 2	NC 3	
8	Si le développement du logiciel est externalisé, un cahier des charges complet est-il formalisé ou des documents équivalents servant à définir la commande pour le développement du logiciel ? Si les logiciels sont développés en interne, un document de spécification est-il disponible ?	Ce cahier des charges doit présenter l'ensemble des exigences issues des spécifications générales.	X	X	X	
9	Testabilité des exigences	Plan de test de la MMRi conforme aux spécifications et couvrant l'ensemble des conditions de déclenchement normales et sur défaut, vérification des temps de réponse.	X	X	X	
VALIDATION GLOBALE						
10	Réalisation de SAT (Tests sur site des fonctions de sécurité)	Les SAT sont réalisées sur site, l'automate étant connecté à ses capteurs et actionneurs. Le but est de réaliser l'ensemble des essais non destructifs de la fonction de sécurité. Les SAT sont en général réalisées pour la MMRi dans son ensemble. L'objectif est de vérifier que la fonction de sécurité est conforme à la spécification générale.	X	X	X	
11	Identification de la version testée	Version logicielle identifiée par un n° de version et/ou un CRC ou un checksum.	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 2-SPÉCIFICATIONS FONCTIONELLES ET VALIDATION GLOBALE						
n°	Questions	Éléments de discussion	Applicable			Explications de l'exploitant et commentaires de l'inspection
			NC 1	NC 2	NC 3	
12	Réalisation de tests fonctionnels	Test des conditions d'activation : logiques de vote, réaction sur les différents seuils d'alarme et de déclenchement. Si possible (si ça ne génère pas de risque), ces tests sont réalisés en générant le phénomène physique au niveau du capteur.	X	X	X	
13	Mesure des temps de réponse	Temps de réponse mesurés avec une précision suffisante pour le temps de réponse attendu.	X	X	X	
14	Réalisation de tests sur défaut	Il s'agit de tests non destructifs : perte d'un capteur, signal hors échelle, pertes d'alimentation…	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 3- CONCEPTION DÉTAILLÉE						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
SPÉCIFICATION LOGICIELLE						
1	Spécification logicielle issue de la spécification système	La spécification du logiciel doit reprendre l'ensemble des exigences système (SRS) applicables et les traduire de manière à pouvoir développer un logiciel applicatif conforme à l'ensemble des exigences système.	X	X	X	
2	Description du fonctionnement : Identification des entrées de l'automate, des variables correspondantes et des logiques de vote	La logique peut être traduite sous forme de logigramme, de grafcet, etc.	X	X	X	
3	Seuils de déclenchement	Traduction des seuils de déclenchement capteur en seuil de déclenchement automate.	X	X	X	
4	Identification de l'action de sécurité	Quel est l'état des sorties correspondant à la position de sécurité ? Selon quelle séquence sont-elles activées ? Y-a-t-il des temporisations ?	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 3- CONCEPTION DÉTAILLÉE						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
5	Identification des valeurs d'entrée en défaut ou hors gamme des capteurs et des actionneurs	Les éléments suivants peuvent être signalés et traités comme des défauts : • Valeurs hors échelle de mesure (valeur haute ou basse) ; • Signal d'équipement en défaut ; • Défauts ou retards de communication pour les réseaux de terrain ; • Défauts d'alimentations des entrées sorties. Le comportement (mise en repli ou alarme doit être identifié). *Rq : Les auto-tests de l'automate sont réalisés par ailleurs et ne font pas partie du logiciel applicatif.*	X	X	X	
6	Prise en compte des contraintes matérielles	La conception du logiciel applicatif doit prendre en compte les contraintes matérielles de l'automate : • Temps de cycle ; • Charge UC ; • Mémoires.		X	X	
7	Conditions de réarmement	Les conditions de réarmement après activation : il ne doit normalement pas être possible de redémarrer en présence d'un défaut.	X	X	X	
8	Inhibition des fonctions de sécurité	Existe-t-il des moyens d'inhibition logiciels, comment sont-ils réalisés, sont-ils limités dans le temps ?	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 3- CONCEPTION DÉTAILLÉE						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
9	Traçabilité des exigences	La traçabilité entre les spécifications système et les spécifications logicielles doit être vérifiée afin de s'assurer de l'exactitude et de la complétude des spécifications.		X	X	
PROGRAMMATION DU LOGICIEL APPLICATIF						
10	Conception modulaire	Le logiciel doit être structuré en modules correspondant à des fonctions simples. Plusieurs architectures sont possibles par exemple : • Plusieurs modules pour chaque fonction de sécurité (entrées / traitement / sorties / gestion des défauts) ; • Des modules communs à plusieurs fonctions de sécurité (en particulier le module de commande des sorties) ; • Un module par fonction de sécurité complète. Il est recommandé de présenter les modules différents sur des pages de programmation différentes.	X	X	X	
11	Langages de programmation	Utilisation de langages à variabilité limitée ou logiciel conforme à la CEI61508.	X	X	X	
12	Utilisation de modules certifiés ou de bibliothèques de fonctions certifiées	Il est préférable d'utiliser des bibliothèques certifiées pour les niveaux de confiance élevés. Si les modules utilisés ne sont pas certifiés, ils doivent être validés en tests unitaires.		X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 3- CONCEPTION DÉTAILLÉE						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
13	Indépendance des fonctions	Deux fonctions intervenant sur un même scénario ne doivent pas manipuler les mêmes variables.		X	X	
14	Existence de bonnes pratiques de programmation	Les bonnes pratiques doivent permettre de garantir la fiabilité et la maintenabilité du logiciel Un document de bonnes pratiques (général à l'installation ou spécifique à un projet) peut présenter : • des règles d'utilisation des blocs fonctionnels ; • des règles de nommage des variables ; • des règles sur la structure du programme ; • des règles sur les commentaires.			X	
TESTS UNITAIRES ET FAT						
15	Réalisation de tests unitaires	Les tests unitaires permettent de valider le bon fonctionnement et le bon paramétrage des blocs fonctionnels élémentaires (vote 1oo2, temporisation, inhibition…). Ils ne sont pas nécessairement tracés mais peuvent faire partie du processus de programmation. Ils ne sont pas nécessaires pour les blocs certifiés.			X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 3- CONCEPTION DÉTAILLÉE						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
16	Plan de tests d'intégration (Plan de FAT) issu de la spécification logicielle	Les plans de tests d'intégration doivent permettre de tester l'ensemble des conditions d'activation des fonctions de sécurité et des conditions de réarmement et les comportements sur défaut. Les plans de tests d'intégration doivent couvrir l'ensemble des combinaisons d'entrées. Il n'existe pas toujours de plans de tests détaillés, il convient dans ce cas de s'assurer que des moyens sont mis en œuvre pour réaliser ces tests dans le cadre du développement des logiciels applicatifs. Quoi qu'il en soit, il semble nécessaire de tracer ces tests pour des NC>=2.		X	X	
17	Réalisation de tests d'intégration (FAT)	Les FAT sont réalisées une fois le logiciel applicatif chargé dans l'automate par simulation des entrées et lecture des sorties. Bien souvent les FAT sont réalisées pour des nouveaux projets et pas pour des modifications sur site. (Cf. gestion des modifications).		X	X	
18	Identification de la version testée	Version logicielle identifiée par un n° de version et/ou un CRC.	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 3- CONCEPTION DÉTAILLÉE						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
19	Réalisation des tests fonctionnels	Toutes les conditions de déclenchement doivent être testées, y compris pour les logiques de vote. Les résultats obtenus doivent être conformes aux résultats attendus et enregistrés. La version testée doit être identifiée.	X	X	X	
20	Réalisation des tests sur défaut	Les défauts identifiés en spécifications générales et détaillées doivent être simulés (valeur hors gamme, capteur en défaut, capteur absent, défaut carte d'entrée, etc.). Les résultats obtenus doivent être conformes aux résultats attendus et enregistrés. La version testée doit être identifiée.	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 4 - SUIVI DU LOGICIEL EN EXPLOITATION						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
GESTION DES VERSIONS						
1	Identification des versions installées	Identification par un numéro de version et une date des versions en cours de validité. Identification du numéro de version de chaque fonction ou de chaque page de programmation. Identification par un CRC ou une signature.	X	X X	X X X	
2	Historique des versions	L'ensemble des versions doivent être identifiables par une date et un statut (en cours de validité, en développement, abandonné).	X	X	X	
3	Information sur les versions	Les modifications réalisées entre les différentes versions doivent être expliquées. Les outils logiciels utilisés pour le développement, document de conception et de validation, personnes intervenues pour la modification et configuration matérielle pour laquelle la version est valable doivent être identifiés.	X	X X	X X	
4	Archivage des versions	L'archivage des versions doit permettre de restaurer au minimum la version antérieure et la version en cours de validité. Elles doivent être archivées dans deux lieux différents (il est recommandé de disposer d'un exemplaire sur site).	X	X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 4 - SUIVI DU LOGICIEL EN EXPLOITATION						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
GESTION DES MODIFICATIONS						
5	Existence d'une procédure de gestion des modifications du logiciel	Toute modification doit faire l'objet d'une demande de modification, identifiant les raisons et analysant l'impact sur la sécurité.	X	X	X	
		L'analyse d'impact doit évaluer l'impact des modifications d'une fonction sur les autres fonctions de sécurité de l'automate.	X	X	X	
		Les personnes responsables d'autoriser et de valider la modification doivent être identifiées.	X	X	X	
6	Archivage de la version courante du logiciel avant modification	La version archivée doit être récupérable.	X	X	X	
7	Cycle de vie des modifications	En fonction des résultats de l'analyse d'impact, les modifications doivent suivre totalement ou en partie le processus de développement et validation que le logiciel initial.	X	X	X	
8	Protection contre les modifications	Seules les personnes autorisées doivent avoir la possibilité de réaliser des modifications.	X	X	X	
		Les logiciels doivent être protégés par un système de verrouillage ou un code d'accès.		X	X	

Fiche d'évaluation détaillée des logiciels applicatifs

GRILLE 4 - SUIVI DU LOGICIEL EN EXPLOITATION						
n°	Questions	Éléments de discussion	Applicable			Applicable
			NC 1	NC 2	NC 3	
9	Validation de la modification	Des essais de réévaluation doivent permettre de s'assurer que la sécurité n'a pas été dégradée. Ils doivent couvrir les phases de tests pertinentes suivant l'importance de la modification. (FAT, SAT, temps de réponse, acquisition d'une mesure). Les plans de tests doivent être cohérents avec les plans de FAT et SAT précédents si des FAT ou SAT sont réalisées en modification.	X	X	X	
Tests périodiques						
10	Réalisation d'un relevé des versions	Vérification que les versions implémentées correspondent aux versions validées. L'interface de programmation des automates doit permettre de relever la version effectivement en cours d'utilisation ainsi que la date de la dernière modification.		X	X	
11	Réalisation de tests fonctionnels et sur défaut	Ces tests correspondent aux tests réalisés en SAT. Ils permettent de vérifier la non régression de la performance de la barrière.	X	X	X	

ANNEXE 4 FICHE D'INSPECTION

Fiche d'inspection des logiciels applicatifs

<table>
<tr><td colspan="2">Société inspectée :</td><td colspan="2">Date :</td></tr>
<tr><td rowspan="3">Thème de la visite :</td><td rowspan="3">MMRi – Développement, validation et maitrise des logiciels applicatifs</td><td>Type de visite d'inspection :</td><td rowspan="2">Pilote de la visite d'inspection :</td></tr>
<tr><td>Secteur industriel :</td></tr>
<tr><td>Type d'installation :</td><td rowspan="2">Autres inspecteurs :</td></tr>
<tr><td colspan="3">Référentiel :</td></tr>
</table>

Fiche d'inspection simplifiée « LOGICIELS APPLICATIFS DES MMRi »

- L'objectif de cette grille est de répondre aux questions suivantes :
  - Q1 : Le développement du logiciel est-il issu d'une spécification fonctionnelle détaillée de la MMRI ?
  - Q2 : Des bonnes pratiques de programmation sont-elles appliquées ?
  - Q3 : Un plan de validation permet-il de vérifier l'ensemble des exigences ?
  - Q4 : S'assure-t-on que les performances du logiciel sont maintenues dans le temps ?
  - Q5 : Les processus relatifs au logiciel permettent-il d'atteindre et maintenir un niveau de sécurité ?
- Cette grille suit les différentes étapes du cycle de vie du logiciel et ses processus support.
- Des points facilement vérifiables sont donnés pour chacune des étapes (cycle de vie, spécification, conception détaillée, tests d'acceptation, gestion de version, gestion de modifications, tests périodiques)
- La grille ne propose pas de liste type des documents à demander, la gestion documentaire étant variable d'une entreprise à une autre. Elle identifie les informations à rechercher aux différentes étapes du cycle de vie. Il faut garder à l'esprit qu'on cherche à évaluer les performances globales d'une MMRi à partir des performances propres au logiciel applicatif.
- La grille indique si les exigences sont applicables pour des FIS NC1, NC2 ou NC3. Cette indication est faite pour ne pas être excessivement pénalisante. Cependant, un processus d'ingénierie des logiciels devrait être mis en place quel que soit le niveau de confiance visé et l'ensemble des exigences devrait être respecté. Si des FIS de niveaux de confiance différents sont traitées par le même automate, l'ensemble du logiciel applicatif doit être réalisé en respectant les exigences du NC le plus élevé.

Fiche d'inspection des logiciels applicatifs

Information à demander dans la lettre d'annonce de l'inspection et à tenir à disposition lors de l'inspection :

- ✓ Identification des MMRi de l'installation
- ✓ Les spécifications des fonctions de sécurité instrumentées ;
- ✓ Si les développements sont externalisés le cahier des charges (ou documents équivalents servant à définir la commande pour le développement des logiciels applicatifs) ;
- ✓ Les règles ou bonnes pratiques de développement des logiciels applicatifs appliquées ;
- ✓ Les plans et résultat de tests de validation d'une fonction de sécurité instrumentée ;
- ✓ Les procédures de gestion des versions et des modifications.

Lors de l'inspection, les personnes compétentes pour les différentes phases du cycle de vie du logiciel devront être présentes :

- ✓ Spécifications fonctionnelles ;
- ✓ Validation des logiciels applicatifs ;
- ✓ Maintenance et modification des automates et logiciels.

Fiche d'inspection des logiciels applicatifs

Q1 : Le développement du logiciel est-il issu d'une spécification fonctionnelle détaillée de la MMRI ?						
N°	**Question**	**Eléments de discussion**	**Applicable**			**Réponse de l'exploitant et commentaires de l'inspection**
			NC 1	**NC2**	**NC3**	
1.1	Présentation des MMRI de l'installation	*On cherchera à avoir une vision de l'existant sur l'installation et à sélectionner un ou deux fonctions pertinentes comme support à l'inspection* : • identification des MMRI de l'installation ; • identification de leur Niveau de confiance ; • identification des types d'équipement utilisés (automates de sécurité, automate de contrôle commande, centrales feu et gaz).	X	X	X	
1.2	Choix d'une ou plusieurs MMRI pour la suite de l'inspection : Quelle est la fonction de sécurité ? Sur quel scénario intervient-elle ?	L'objectif est de partir d'une spécification de très haut niveau de la fonction dans son ensemble. On commence par identifier les MMRi intervenant sur un scénario et leur NC. *Rq : Si plusieurs fonctions d'un même scénario sont traitées par le même automate les règles d'indépendance devront être vérifiées.*	X	X	X	

1.3	Existe-t-il une description fonctionnelle de la MMRI sélectionnée pouvant servir de base à la spécification logicielle et à un plan de validation ?	Identification des capteurs et des points de mesure ; Identification des actionneurs ; Identification des types d'entrées / sorties (analogique, numérique, TOR) ; Description de l'action de sécurité et des conditions de déclenchement ; Description des comportements sur défauts (passage en repli ou déclenchement d'alarme) à prendre en compte dans la programmation (défauts capteurs, perte de signaux, …) ; Exigences de temps de réponse ; Conditions d'inhibition.	X	X	X	
1.4	Les spécifications fonctionnelles sont-elles transmises aux entités en charge du développement ?	Les spécifications peuvent être transmises sous forme de cahier des charges, document de spécification plus ou moins détaillées, SRS. L'acceptation de la fourniture devra être faite à partir de la vérification du respect de ces spécifications.	X	X	X	

Fiche d'inspection des logiciels applicatifs

Q2 : L'application de bonnes pratiques de développement est-elle demandée ?						
N°	**Question**	**Eléments de discussion**	**Applicable**			**Réponse de l'exploitant et commentaires de l'inspection**
			NC1	**NC2**	**NC3**	
2.1	Existe-t-il une spécification logicielle détaillée issue de la spécification fonctionnelle ?	Cette spécification doit reprendre l'ensemble des exigences fonctionnelles et les traduire en données exploitables pour la programmation : (identification des entrées sorties, des seuils, etc.). Le fonctionnement du logiciel doit être décrit, sous forme de logigramme par exemple.		X	X	
2.2	Le niveau de sécurité souhaité est-il spécifié ?	Le responsable du développement du logiciel doit être informé des exigences de sécurité.	X	X	X	
2.3	Y-a-t-il des exigences sur les bonnes pratiques de programmation ?	Des pratiques de programmation permettant d'atteindre les niveaux de sécurité souhaités doivent être définis. Il faut par exemple : • traiter les valeurs hors gamme ; • se limiter à l'utilisation des bibliothèques certifiées pour les automates de sécurité ; • des règles de nommage ou de présentation peuvent faciliter la relecture du code.		X	X	
2.4	Les exigences de validation sont-elles spécifiées ?	Un document doit décrire le type de preuves attendues par l'utilisateur pour la validation du logiciel.	X	X	X	

Q3 Un plan de validation permet-il de vérifier que toutes les exigences sont appliquées ?						
N°	**Question**	**Eléments de discussion**	**Applicable**			**Réponse de l'exploitant et commentaires de l'inspection**
			NC1	**NC2**	**NC3**	
3.1	La réalisation de tests unitaires par le fournisseur est-elle requise ?	Les tests unitaires permettent de vérifier unitairement chaque bloc fonctionnel élémentaire. Il s'agit d'une bonne pratique de programmation, des tests sont normalement réalisés par le développeur du logiciel.			X	
3.2	Des tests d'intégrations (ou tests d'acceptation en usine : FAT) ont-ils été prévus et réalisés ?	Les FAT sont réalisées sur une version du logiciel complète et figée et sur un matériel (API) identique à celui installé sur site (y compris l'OS). Un plan de FAT doit couvrir l'ensemble des exigences de spécification détaillée. Il faut en particulier assurer la couverture des différentes conditions de déclenchement, les logiques de vote, les tests aux limites, les tests de comportement sur défauts. Pour des fonctions simples, entièrement testable sur site, il peut être possible de réaliser ces tests en SAT. Il faut cependant faire attention à ce que ces tests soient bien complets. Pour des NC2 ou NC3, il est recommandé que l'utilisateur assiste aux SAT.	X	X	X	

3.3	Des tests d'acceptation sur site (SAT) ont-ils été réalisés ?	Les SAT sont réalisées sur site, l'automate étant connecté à ses capteurs et actionneurs. Le but est de réaliser l'ensemble des essais non destructifs de la fonction de sécurité. Les SAT sont en général réalisées pour la MMRi dans son ensemble L'objectif est de vérifier que la fonction de sécurité est conforme à la spécification fonctionnelle. La SAT est moins complète que la FAT, on ne couvre pas toutes les conditions de déclenchement et les comportements sur défauts. On doit néanmoins s'assurer du branchement correct de tous les équipements (vérification des liaisons), du déclenchement de la fonction et des temps de réponse L'utilisateur et le concepteur doivent assister aux SAT.	X	X	X	
3.4	Les différentes phases de tests sont-elles documentées ?	Les plans de tests doivent couvrir l'ensemble des exigences fonctionnelles et des comportements sur défauts (la majorité de ces cas sont traités en FAT). Les moyens et résultats attendus doivent être précisés. Les résultats doivent être consignés dans des fiches de tests. En cas de résultat non conformes, une analyse et des corrections doivent être enregistrées. Les versions logicielles validées doivent être données dans la documentation.	X	X	X	

Fiche d'inspection des logiciels applicatifs

Q4 : S'assure-t-on que les performances du logiciel sont maintenues dans le temps ?						
N°	**Question**	**Eléments de discussion**	**Applicable**			**Réponse de l'exploitant et commentaires de l'inspection**
			NC1	**NC2**	**NC3**	
4.1	Y-a-t-il une gestion des versions ?	Les versions en cours de validité doivent être identifiées et archivées. On doit être en mesure de relever la version du logiciel intégrée à l'automate pour la comparer à la version validée. L'identification peut être faite par un n° de version, une somme de contrôle et/ou une date.	X	X	X	
4.2	Y-a-t-il une gestion des modifications ?	Les modifications doivent être réalisées suivant une procédure et donner lieu à des enregistrements. Une analyse d'impact doit évaluer l'impact des modifications d'une fonction sur les autres fonctions de sécurité de l'automate. Les personnes habilitées à autoriser et valider les modifications doivent être identifiées. Une validation (tests) doit être faite et enregistrée suite aux modifications.	X	X	X	
4.3	Le système est-il verrouillé contre les modifications ?	Seules les personnes autorisées doivent avoir la possibilité de réaliser des modifications. La protection peut être faite par plusieurs niveaux de codes d'accès.	X	X	X	
4.4	Des tests périodiques sont-ils réalisés ?	Ces tests fonctionnels et sur certains défauts (type absence capteur) correspondant à ceux réalisés en SAT. Ils permettent de vérifier la non régression de la performance de la barrière.	X	X	X	

4.5	Un relevé de version est-il réalisé ?	Lors du test périodique, la version logicielle effectivement installée est-elle relevée et comparée avec la version théoriquement installée ?		X	X	

Q5 : S'assure-t-on que les performances du logiciel sont maintenues dans le temps ?						
N°	**Question**	**Eléments de discussion**	**Applicable**			**Réponse de l'exploitant et commentaires de l'inspection**
			NC1	**NC2**	**NC3**	
5.1	Y-a-t-il une description du cycle de vie du logiciel ?	*Le cycle de vie doit faire apparaitre des étapes de spécification, conception, validation et des processus supports pour la gestion des versions et des modifications ainsi que pour les tests.*		X	X	
5.2	Les personnes ou entités responsables des différentes phases sont-elles identifiées ? Leurs compétences sont-elles vérifiées ? Leur niveau d'indépendance est-il suffisant ?	Entités ou personnes responsables de la maitrise de différentes phases du cycle de vie (spécification, réalisation, validation, maintenance, modification). Personnes et entités impliquées dans la réalisation des différentes phases (sous-traitance, compétence interne). Compétence sur les technologies mises en œuvre (équipements et outils de programmation utilisés). Les responsables de la validation et de la réalisation doivent être des personnes différentes. En particulier les personnes qui réalisent les FAT et les SAT doivent être indépendantes des personnes qui réalisent le développement. Au minimum, les niveaux d'indépendance suivants sont requis : • pour un niveau NC1, une personne indépendante réalise les tests ; • pour un NC2, un service indépendant est chargé des tests ; • pour un NC3 une organisation indépendante (organisme tiers) valide les tests.		X	X	

ANNEXE 5 LISTE DES DOCUMENTS TYPES POUR LA PRÉPARATION ET LA RÉALISATION DE L'INSPECTION

Documents types pour l'inspection des l'inspection du cycle de vie des logiciels applicatifs

Thèmes de l'inspection	Listes et contenu des documents
Contexte et organisation	
• Identification des MMRI	➢ Listes des MMRI ou nœuds papillons ➢ Certification des MMRI ➢ Rex sur les MMRI
• Organisation pour le développement et la validation des logiciels	➢ Processus ou plan qualité relatif aux logiciels applicatifs ou à un projet ➢ Cahier des charges pour la sous-traitance ➢ Grilles de compétence
Spécifications et validation globales	
• Spécifications fonctionnelles	➢ SRS ou spécification ➢ Cahier des charges pour la sous-traitance ➢ Plan de test de validation
• Tests d'acceptation (SAT)	➢ Fiches de tests renseignées
Conception détaillée	➢
• Spécification logicielle	➢ Spécification logicielle ➢ Logigramme ou algorithme ➢ Plan de test d'intégration
• Programmation du logiciel	➢ Règles de programmation ➢ Identification des modules utilisés ➢ Identification des entrées sorties ➢ Logiciel commenté
• Tests d'intégration	➢ Fiches de tests renseignées

Thèmes de l'inspection	Listes et contenu des documents
Suivi en exploitation • Gestion des versions • Gestion des modifications • Tests périodiques	➢ ➢ Procédure des gestions de versions ➢ Archivage des différentes versions ➢ Procédure de gestion des modifications ➢ Analyses d'impact ➢ Documents de validation de modifications ➢ Compte rendu de relevé de versions ➢ Plans de tests périodiques ➢ Fiche de tests périodiques renseignée

ANNEXE 6 PROPOSITION DE PLAN D'INSPECTION

Type de visite d'inspection : Approfondie	**Pilote de la visite d'inspection :**
Secteur industriel :	
Type d'installation :	**Autres inspecteurs :**
Thème de la visite : MMRi – Développement, validation et maitrise des logiciels applicatifs	

Thèmes d'inspection	Durée	Evaluation / Assessment	Personnes
Présentation des objectifs	30 minutes	**Objet :** Prise en compte des logiciels applicatifs dans la mise en œuvre et la validation des MMRI **Base :** Grilles d'évaluation ✓ CEI 61511 ✓ Doctrine MMRI Observations et questions de l'industriel	
Contexte et organisation	1h30	**Identification de l'existant sur le site :** technologies, antériorité **Cycle de vie mis en œuvre :** processus, organisation, responsabilités, intervenants internes et externes	
Spécification et validation globales	2h	**Spécifications générales :** Spécification détaillée de la MMRI servant de point d'entrée à la réalisation du logiciel **Validation** Tests Acceptation finale du logiciel	
Conception détaillée	2h	**Spécifications logicielles** Traçabilité par rapport aux specs générales Intégration d'exigences techniques spécifiques pour atteindre le NC **Réalisation :** Bonnes pratiques de développement Vérification des règles essentielles de conception **Vérification** Tests unitaires et tests d'intégration (FAT)	
Suivi en Exploitation	1h30	**Gestion des versions** **Gestion des modifications** **Tests périodiques**	
Bilan	45 minutes	Préparation du bilan Retour de l'industriel Retour de l'inspecteur	

INERIS

maîtriser le risque
pour un développement durable

Institut national de l'environnement industriel et des risques

Parc Technologique Alata
BP 2 - 60550 Verneuil-en-Halatte

Tél. : +33 (0)3 44 55 66 77 - Fax : +33 (0)3 44 55 66 99

E-mail : ineris@ineris.fr - **Internet :** http://www.ineris.fr