



CENTRE NATIONAL D'ÉTUDES SPATIALES

REFERENTIEL NORMATIF du CNES RNC

Référence : **RNC-CNES-Q-ST-80-100**
Version 4
02 Juin 2008

STANDARD

ASSURANCE PRODUIT EXIGENCES SIMPLIFIEES D'ASSURANCE QUALITE POUR LE DEVELOPPEMENT DES LOGICIELS

ACCORD du Bureau de Normalisation	BN n° 44 du 08/08/09
APPROBATION Président du CDN Alain CUQUEL	

PAGE D'ANALYSE DOCUMENTAIRE

TITRE : EXIGENCES SIMPLIFIEES D'ASSURANCE QUALITE POUR LE DEVELOPPEMENT DES LOGICIELS	
MOTS CLES : Qualité logiciel - Développement logiciels – R&T	
NORME EQUIVALENTE : Néant	
OBSERVATIONS : Néant	
<p>RESUME : Ce document définit les exigences simplifiées d'Assurance Qualité des Logiciels applicables :</p> <ul style="list-style-type: none"> ✍ au développement et à la maintenance des systèmes informatiques sol et des logiciels embarqués par des PME ou des Laboratoires scientifiques. ✍ aux développements informatiques liés à la Recherche et Technologie. 	
<p>SITUATION DU DOCUMENT : Ce document fait partie de la collection des standards associés au Référentiel Normatif du CNES. Il est affilié au document « RNC-ECSS-Q-ST- 80 Software Product Assurance ».</p>	
NOMBRE DE PAGES : 45	LANGUE : Française
Progiciels utilisés / version : Word 2002	
SERVICE GESTIONNAIRE : Inspection Générale Direction de la Fonction qualité (IGQ)	
<p>AUTEUR(S) :</p> <p>Initialement établi par G. GUILLOT/J.F. POBLE repris par J-C. DAMERY</p>	<p>DATE : 02/06/2008</p>

© CNES 2008

Reproduction strictement réservée à l'usage privé du copiste, non destinée à une utilisation collective (article 41-2 de la loi n°57-298 du 11 Mars 1957).

PAGES DES MODIFICATIONS

VERSION	DATE	PAGES MODIFIEES	OBSERVATIONS
PR.0	26/08/93		Création
1.0	21/04/94	i.1 ; i.2 ; i.3 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6	Approbation du Comité Technique Référentiel et du Comité de Validation
2	01/03/00		Nouvelle codification des documents (ancienne référence « MPM-53-00-09 »)
3	08/12/03	Toutes	Restructuration du document ✍ pour mise en conformité avec la norme RNC-ECSS-Q-80A et ajout de plans-types ✍ pour élargir la portée du document au cas des développements logiciels dans le domaine de la R&T
4	02/06/2008	Toutes	Changement de nomenclature suite à la phase de benchmarking ECSS (ancienne référence RNC-CNES-Q-80-509).

SOMMAIRE

1	INTRODUCTION	7
2	OBJET	7
3	DOMAINE D'APPLICATION	7
4	GUIDE DE LECTURE ET D'APPLICATION.....	7
4.1	MODULATION DES EXIGENCES SELON LE CONTEXTE DU PROJET	7
4.2	PRECISIONS SUR LE CONTEXTE DE R&T.....	8
5	DOCUMENTS ASSOCIES	8
5.1	DOCUMENTS DE REFERENCE	8
5.2	DOCUMENTS APPLICABLES	9
6	SIGLES ET ABREVIATIONS.....	10
7	PRINCIPES FONDAMENTAUX DE LA METHODE.....	11
8	ORGANISATION ET ASSURANCE PRODUIT.....	11
9	DESCRIPTION DE LA DEMARCHE	12
9.1	CYCLE DE DEVELOPPEMENT	12
9.1.1	Spécification du logiciel.....	13
9.1.2	Conception	14
9.1.3	Codage - Tests unitaires.....	14
9.1.4	Validation.....	15
9.1.5	Recette	17
9.2	GESTION DE LA CONFIGURATION – GESTION DES MODIFICATIONS	17
9.2.1	Principes	18
9.2.2	Cas des applications UNIX.....	19
9.2.3	Document descriptif de la configuration.....	19
9.3	METHODES, OUTILS, REGLES.....	19
9.3.1	Principes	19
9.3.2	Règles de nomenclature	20
9.3.3	Règles de codage.....	20
9.4	CONDITIONS PARTICULIERES DE DEVELOPPEMENT	21

9.4.1	Sûreté de fonctionnement et sécurité des logiciels	21
9.4.2	Logiciels réutilisés.....	22
9.4.3	Interface Homme – Machine.....	23
9.5	DOCUMENTATION PRECONISEE DANS LE CAS D'UN DEVELOPPEMENT LOGICIEL STANDARD....	24
9.6	DOCUMENTATION PRECONISEE DANS LE CAS D'UN DEVELOPPEMENT LOGICIEL DE R&T	25
10	ANNEXE 1 : PLANS-TYPE.....	26
10.1	PLAN D'APPLICATION.....	26
10.1.1	Présentation	26
10.1.2	Plan-type du plan d'application.....	26
10.2	DOCUMENT DE SPECIFICATION	28
10.2.1	Présentation	28
10.2.2	Plan-type du document de spécification	28
10.3	DOCUMENT DE CONCEPTION	30
10.3.1	Présentation	30
10.3.2	Plan-type du document de conception.....	30
10.4	DOCUMENT DE VALIDATION	31
10.4.1	Présentation	31
10.4.2	Plan-type du document de validation.....	31
10.5	MANUEL D'UTILISATION.....	34
10.5.1	Présentation	34
10.5.2	Plan-type du manuel d'utilisation.....	34
10.6	CAHIER DE RECETTE.....	36
10.6.1	Présentation	36
10.6.2	Plan-type du cahier de recette	36
11	ANNEXE 2 : INDEX DES REGLES	39
11.1	CAS D'UN DEVELOPPEMENT LOGICIEL STANDARD.....	39
11.2	CAS D'UN DEVELOPPEMENT LOGICIEL CRITIQUE	41
11.3	CAS D'UN DEVELOPPEMENT LOGICIEL R&T.....	43

1 INTRODUCTION

Ce document "Exigences simplifiées d'Assurance Qualité pour le développement des logiciels" fait partie de la collection des standards associés au document « RNC-ECSS-Q-ST-80 Software Product Assurance ».

Il est également inclus dans le « Guide pour les Projets Scientifiques » disponible sur le site Internet GNS (Guide Normatif Simplifié - <http://gns.cnes.fr>).

2 OBJET

Le présent document a pour but de présenter des exigences qualité logiciel simplifiées permettant d'obtenir un niveau de qualité logiciel satisfaisant pour les logiciels d'application développés dans le cadre de certains projets du CNES, par les industriels fournisseurs, par les laboratoires scientifiques ou en interne au CNES, ainsi que ceux développés dans un contexte de Recherche et Technologie (R&T dans la suite de ce document).

3 DOMAINE D'APPLICATION

Ce document est applicable aux développements de logiciels destinés à des applications pour lesquelles on accepte un « relâchement » des exigences qualité telles qu'énoncées dans l'ECSS-Q-ST-80, en relation avec le niveau de risque acceptable.

Ce document s'applique également au cas des développements de logiciels dans un contexte de R&T pour lesquels le niveau qualité ne revêt pas d'importance majeure.

Il sert :

- ? dès la phase A du projet pour définir le cadre de développement aux futurs réalisateurs,
- ? dès le début des développements (fin de phase B - début de phase C) comme base de discussions avec le réalisateur de logiciels afin de finaliser les dispositions pratiques.

Ces dispositions sont définies lors de réunions de travail entre le CNES et ses fournisseurs ; elles sont formalisées dans la proposition du fournisseur et éventuellement reportées dans un plan d'application (cf. EX1).

4 GUIDE DE LECTURE ET D'APPLICATION

Le présent document est structuré en exigences (de nomenclature EX n) et en recommandations (de nomenclature Recom j) applicables à un développement logiciel.

Certaines des exigences sont modulées selon le contexte du développement logiciel :

- ? Criticité de ce logiciel,
- ? Taille du logiciel,
- ? Contraintes de coût et de délai,
- ? Contexte R&T de développement informatique.

A chaque exigence est associé un contexte d'application.

4.1 MODULATION DES EXIGENCES SELON LE CONTEXTE DU PROJET

Il appartient au CNES de se prononcer sur la modulation des exigences et ainsi de déterminer le lot d'exigences et de recommandations applicables à un développement logiciel particulier, et donc de demander au fournisseur en charge de ce développement d'indiquer la façon dont il entend répondre aux exigences et recommandations.

Une classification est proposée dans ce document pour traiter les deux cas de logiciels « **hors R&T** » et de logiciels « dans un **contexte R&T** ». Dans le contexte « hors R&T », le cas des logiciels un peu plus **critiques** que le cas **standard** introduit quelques exigences supplémentaires.

La modulation des exigences pour un projet peut conduire à opérer une autre classification, comme par exemple l'application de tout ou partie des exigences de R&T pour un projet de très petite taille.

4.2 PRECISIONS SUR LE CONTEXTE DE R&T

Il est important de noter qu'il faut dissocier les **développements logiciels de R&T** de **l'industrialisation** qui est faite pour certains logiciels à l'issue de ces développements, mais hors contexte R&T.

Dans cette optique, la classification des exigences « R&T » proposée dans ce document s'adresse à la première étape de développement de R&T et ne traite donc pas de l'industrialisation.

Pour traiter l'étape d'industrialisation, les exigences d'assurance produit doivent être complétées afin d'inclure les actions de fiabilisation du produit sur les aspects suivants :

- ? Précisions sur la documentation
- ? Environnement d'essai (documentation, jeux de tests)
- ? Fiabilisation du code (voire recodage complet en phase d'industrialisation), en termes d'interface, de gestion des entrées,
- ? Stratégie de gestion et de remontée d'erreurs.

5 DOCUMENTS ASSOCIES

5.1 DOCUMENTS DE REFERENCE

- DR1 : Software Product Assurance
RNC-ECSS-Q-ST-80
- DR2 : Software general requirements
RNC-ECSS-E-ST-40
- DR3 : Contenu d'une spécification technique de besoin logiciel
RNC-CNES-E-HB-40-501
- DR4 : Bilan Technique
RNC-CNES-Q-ST-20-100
- DR5 : Commission de Revue des Essais
RNC-CNES-Q-ST-20-101
- DR6 : Contenu d'un dossier des logiciels réutilisés
RNC-CNES-Q-HB-80-514
- DR7 : Règles pour l'utilisation du langage FORTRAN 77
RNC-CNES-Q-HB-80-505

- DR8 : Règles pour l'utilisation du langage C
RNC-CNES-Q-HB-80-506
et son annexe pour les logiciels embarqués - RNC-CNES-Q-HB-80-506-A
- DR9 : Règles pour l'utilisation du langage Ada
RNC-CNES-Q-HB-80-504
et son annexe pour Ada 95 - RNC-CNES-Q-HB-80-504-A
- DR10 : Règles pour l'utilisation du langage C++
RNC-CNES-Q-HB-80-513
- DR11 : Règles pour l'utilisation du langage FORTRAN 90
RNC-CNES-Q-HB-80-517
- DR12 : Règles pour l'utilisation du langage Ada dans les logiciels embarqués
RNC-CNES-Q-HB-80-528
- DR13 : Guide pour la sélection et l'interprétation des mesures de complexité du logiciel
RNC-CNES-Q-HB-80-503
- DR14 : Manuel utilisateur et manuels d'exploitation d'un système informatique sol de traitement
RNC-CNES-E-HB-40-503
- DR15 : Règles et recommandations pour conduire une recette de produit logiciel
RNC-CNES-E-HB-40-506
- DR16 : Dossier Descriptif de la Configuration d'un logiciel
RNC-CNES-M-HB-40-516
- DR17 : Exigences standards de gestion de configuration des logiciels
RNC-CNES-M-ST-40-100
- DR18 : Règles pour l'utilisation du langage Java
RNC-CNES-Q-HB-80-527
- DR19 : Règles pour l'utilisation du langage Wave
RNC-CNES-Q-HB-80-521
- DR20 : Règles et Recommandations d'ergonomie des Interfaces Homme Machine informatiques
RNC-CNES-E-HB-40-504
et son annexe guide de vérification : RNC-CNES-E-HB-40-504-A
- DR21 : Règles pour l'utilisation du langage PERL
RNC-CNES-Q-HB-80-533
- DR22 : Règles pour l'utilisation du langage IDL
RNC-CNES-Q-HB-80-534
- DR23 : Règles communes pour l'utilisation des langages de programmation
RNC-CNES-Q-HB-80-501

5.2 DOCUMENTS APPLICABLES

Néant

6 SIGLES ET ABREVIATIONS

BT	Bilan Technique
CRE	Commission de Revue des Essais
Crit.	Applicable uniquement aux logiciels critiques (hors R&T)
DDC	Dossier Descriptif de la Configuration
DR	Document de référence
ECSS	European Cooperation for Space Standardization
RNC	Référentiel Normatif CNES
RSL	Revue de Spécification du Logiciel
R&T	Recherche et Technologie
Std.	Applicable en standard aux logiciels critiques ou non (hors R&T)

7 PRINCIPES FONDAMENTAUX DE LA METHODE

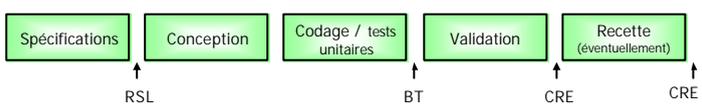
Référence	Libellé	Hors R&T	Contexte R&T
EX1	Tout développement de logiciel doit être mené selon des règles pré-établies, en réponse aux exigences du présent document.	Std	x
EX2	La réponse à ces exigences est reportée dans un plan d'application (dont un plan-type est donné en annexe 1 § 10.1), qui traduit l'engagement du fournisseur sur la prise en compte des exigences décrites ci-après.	Std	

8 ORGANISATION ET ASSURANCE PRODUIT

Référence	Libellé	Hors R&T	Contexte R&T
EX3	Le fournisseur doit préciser l'organisation et les responsabilités de l'équipe en charge du développement, en particulier identifier la personne responsable de l'assurance qualité logiciel.	Std	x
EX4	Le fournisseur doit préciser les formations nécessaires à entreprendre pour combler les manques éventuels dans les compétences disponibles dans l'équipe de développement.	Std	x
EX5	Le fournisseur doit préciser les dispositions d'Assurance Produit qu'il met en place ainsi que la planification des activités correspondantes.	Std	x
EX6	Les dispositions d'Assurance Produit comprennent au minimum la gestion de configuration et la gestion des faits techniques.	Std	x
EX7	Les dispositions d'Assurance Produit doivent être complétées par des dispositions de contrôle et de rédaction de bilans qualité	Std	
EX8	Dans le cas où la prestation consiste à réaliser un système , le fournisseur doit inclure dans sa proposition une identification des risques techniques issus d'une analyse système et leur implication sur le plan de développement du logiciel.	Std	x
EX9	Dans le cas où la prestation consiste à réaliser un sous-système , le fournisseur doit inclure dans sa proposition l'analyse d'impact des risques système identifiés par le CNES.	Std	x
EX10	Le fournisseur doit s'assurer que ses sous-traitants mettent en œuvre des dispositions d'assurance qualité équivalentes. Pour ce faire, il répercute sur ses sous-traitants tout ou partie des exigences qui lui sont imposées.	Std	x

9 DESCRIPTION DE LA DEMARCHE

9.1 CYCLE DE DEVELOPPEMENT

Référence	Libellé	Hors R&T	Contexte R&T
EX11	Le cycle de développement du logiciel doit être défini en tenant compte des caractéristiques techniques du logiciel et des contraintes de développement associées comme le choix des techniques employées (système de gestion de bases de données, réutilisation de produits existants, générateurs d'interface homme/machine, etc.) et les risques inhérents au projet.	Std	x
EX12	Une analyse de besoins doit être menée avant tout développement de logiciel.	Std	
EX13	L'analyse de besoins est menée conjointement par le prestataire et le CNES		x
EX14	Après la phase d'analyse des besoins, le cycle de développement du logiciel doit comporter les phases suivantes : <ul style="list-style-type: none"> - spécification, - conception, - codage - tests unitaires, - validation, - éventuellement recette.  <pre> graph LR RSL --> S[Spécifications] S --> C[Conception] C --> CU[Codage / tests unitaires] CU --> V[Validation] V --> RE[Recette (éventuellement)] BT --> V CRE --> V CRE --> RE </pre>	Std	
EX15	Le cycle de développement peut être itératif : dans ce cas, les itérations sont décrites dans la proposition du fournisseur selon les critères suivants : <ul style="list-style-type: none"> ? Objectifs, ? Activités prévues (qui peuvent reprendre tout ou partie des activités prévues dans les phases classiques du cycle de vie (exigences des paragraphes 9.1.1 à 9.1.5), ? Critères d'achèvement. Il est à noter que les itérations peuvent être affinées au fur et à mesure de l'avancement du développement ; le CNES et le fournisseur précisent les conditions de réalisation de l'itération n+1 à la fin de l'itération n.		x
EX16	Chaque phase doit se terminer par une réunion technique avec le CNES (jalon) pour examiner le travail effectué dans cette phase et le niveau de préparation de la phase suivante.	Std	x

EX17	Dans le cas d'un logiciel intégré dans un équipement , le logiciel participe en tant que constituant à la phase d'intégration matériel-logiciel et son cycle de développement doit être cohérent avec celui du matériel	Crit.	
-------------	--	-------	--

9.1.1 Spécification du logiciel

Référence	Libellé	Hors R&T	Contexte R&T
EX18	<p>Les activités de la phase de spécifications consistent au minimum à :</p> <ul style="list-style-type: none"> ? identifier les besoins exprimés par le client à l'issue de son analyse de besoins puis les traduire en terme de fonctions à remplir par le logiciel, d'interfaces avec l'extérieur et entre elles, ? analyser, en fonction des besoins à couvrir, les logiciels qui pourraient être réutilisés et évaluer les impacts de leur réutilisation sur le développement (cf. § 9.4.2), ? élaborer une version préliminaire du plan de validation. <p>Les éléments en sortie sont décrits dans la documentation du projet (§ 9.5 et 9.6).</p>	Std	x
EX19	<p>Les activités de la phase de spécifications sont complétées par :</p> <ul style="list-style-type: none"> ? des précisions sur les enchaînements de fonctions, ? des précisions sur les contraintes (performances, priorités, encombrement mémoire), 	Std	
EX20	Les éléments en sortie de la phase de spécification sont décrits dans le document de spécifications du logiciel (un plan-type du document est donné en annexe 1 § 10.2).	Std	
EX21	Dans le cas particulier d'un développement comprenant une IHM , la phase de spécifications doit permettre d'élaborer une maquette de l'Interface Homme-Machine (cf § 9.4.3).	Std	x
EX22	Dans le cas particulier d'un développement soumis à des contraintes de sûreté de fonctionnement , la phase de spécifications permet de déterminer les parties critiques du logiciel après une analyse fonctionnelle (cf § 9.4.1).	Crit.	
EX23	Cette phase s'achève par une réunion technique (ou revue) de spécifications du logiciel (RSL) permettant de statuer sur les documents issus de cette phase et sur la réutilisation éventuelle de logiciels. Cette réunion technique doit se dérouler avec la participation des responsables techniques, des responsables assurance produit et/ou des utilisateurs.	Std	

9.1.2 Conception

Référence	Libellé	Hors R&T	Contexte R&T
EX24	L'activité de conception consiste au minimum à : ? définir le découpage structurel de l'application en constituants logiciels puis à détailler chacun d'eux, ? définir le flux de données et les interfaces, Les éléments en sortie sont documentés dans la documentation du projet (§ 9.5 et 9.6).	Std	x
EX25	L'activité de conception est complétée par des estimations de ressources (mémoire centrale, mémoire de masse, CPU, organes périphériques, ...).	Crit.	
EX26	Les éléments en sortie de la phase de conception sont décrits dans le document de conception (un plan-type du document est donné en annexe 1 § 10.3).	Std	
EX27	La conception du logiciel fait apparaître dans des éléments de conception différenciés les fonctions que l'on souhaite réutiliser de façon à pouvoir les valider de manière indépendante.	Std	x

9.1.3 Codage - Tests unitaires

Référence	Libellé	Hors R&T	Contexte R&T
EX28	Les activités de codage et de tests unitaires sont : ? le codage des constituants, ? l'exécution des tests unitaires des constituants et des tests d'intégration entre constituants.	Std	x
Recom 1	Les constituants logiciels sont codés dans un langage de programmation évolué plutôt qu'en assembleur. Le même langage est privilégié pour la réalisation de l'ensemble du logiciel (à l'exception des scripts d'installation).	Std	x
EX29	Le codage respecte les règles énoncées dans le paragraphe 9.3.3.	Std	x

9.1.4 Validation

Référence	Libellé	Hors R&T	Contexte R&T
EX30	Les activités de la phase de validation consistent à dérouler les plans de tests, afin de vérifier que le logiciel remplit ses fonctions spécifiées.	Std	x
EX31	Les essais de validation doivent couvrir toutes les exigences de la spécification du logiciel (un plan-type du document de validation est donné en annexe 1 § 10.4)..	Std	
EX32	Dans le cas d'un développement itératif , les objectifs de validation pouvant varier d'une itération à une autre, les objectifs de validation de l'itération n+1 doivent être définis entre le CNES et le fournisseur à la fin de l'itération n.		x
EX33	Dans le cas d'un logiciel intégré dans un équipement , la validation doit comporter des essais sur le modèle d'identification du matériel correspondant.	Crit.	
EX34	Les essais de validation doivent être réalisés sur un logiciel stable, dont la version est gérée en configuration. Toute modification de la configuration au cours de la phase de validation doit être tracée.	Std	x

Référence	Libellé	Hors R&T	Contexte R&T
EX35	<p>La phase de validation est précédée par un bilan technique (BT) et clôturée par une réunion de fin d'essais (CRE Commission de Revue des Essais) à l'issue de laquelle l'ensemble de la documentation mise à jour doit être acceptée par le CNES.</p> <p>1. Le bilan technique (BT), en début de phase de validation du logiciel, auquel participent le client et le fournisseur, a pour objectif d'autoriser le démarrage des essais de validation. Il permet de :</p> <ul style="list-style-type: none"> - vérifier l'état des actions, - vérifier la complétude et la cohérence du plan et des procédures de tests, - identifier l'état de la configuration et figer la version du logiciel correspondante. <p>A l'issue du BT, si l'autorisation de démarrage des essais est donnée, l'ensemble des procédures de test est déroulé.</p> <p>2. Durant les essais, les résultats de chaque essai sont consignés dans un journal de bord, ainsi que les anomalies et les modifications qui sont identifiées.</p> <p>Dans la mesure du possible, les actions correctives ou évolutives associées aux anomalies et aux modifications identifiées ne sont effectuées qu'à la fin de la phase de validation et donneront lieu à une nouvelle version du logiciel.</p> <p>Dans le cas contraire, toute évolution de la configuration au cours des essais est consignée dans le journal de bord.</p> <p>3. La réunion de fin d'essais (CRE), clôturant la phase de validation, permet d'établir :</p> <ul style="list-style-type: none"> - une synthèse des résultats de tests, - un état des anomalies et des modifications résiduelles, - un état des actions en précisant pour celles qui sont ouvertes leur date d'échéance et leur(s) responsable(s), - l'acceptation ou le refus de la fin de la phase de validation. <p>En cas de refus ou d'acceptation avec réserve(s), les actions à réaliser et leur date limite de réalisation apparaîtront explicitement dans le compte-rendu.</p>	Std	

Référence	Libellé	Hors R&T	Contexte R&T
EX36	<p>La phase de validation est encadrée par 2 réunions.</p> <p>La première permet d'identifier l'état de la configuration et de figer la version du logiciel correspondante.</p> <p>La seconde permet d'établir :</p> <ul style="list-style-type: none"> ? une synthèse des résultats de tests, ? un état des anomalies et des modifications résiduelles, ? l'acceptation ou le refus de la fin de la phase de validation. <p>Durant les essais, les résultats de chaque essai sont consignés, ainsi que les anomalies et les modifications identifiées.</p>		x

9.1.5 Recette

Référence	Libellé	Hors R&T	Contexte R&T
EX37	<p>Dans le cas où il existe une recette du logiciel, celle-ci doit satisfaire aux exigences définies ci-dessus pour la phase de validation du logiciel. La recette s'appuie sur le cahier de recette (éventuellement extrait du document de validation du logiciel – plan-type en annexe 1 § 10.6) et a lieu après acceptation de la phase de validation. Elle se termine par une CRE au cours de laquelle sera prononcée ou non l'acceptation du logiciel développé.</p>	Std	x
EX38	<p>Dans le cas d'un logiciel intégré dans un équipement, la recette est réalisée sur le modèle de qualification du matériel correspondant, ou à défaut sur le modèle le plus représentatif.</p>	Crit.	

9.2 GESTION DE LA CONFIGURATION – GESTION DES MODIFICATIONS

La gestion de la configuration du logiciel correspond à l'ensemble des activités, manuelles ou automatiques, permettant d'identifier, à tout instant, les éléments créés, utilisés ou modifiés par le processus de développement du logiciel, et leurs relations.

Son but principal est de mémoriser chaque version de référence d'un élément, afin de connaître précisément à chaque instant la version utilisée et afin de pouvoir reconstituer, en cas de besoin, toute version antérieure.

9.2.1 Principes

Référence	Libellé	Hors R&T	Contexte R&T
EX39	La gestion de configuration du logiciel doit identifier, de manière unique selon les règles de nomenclature du projet, chaque élément à gérer.	Std	x
EX40	<p>Un membre de l'équipe projet est désigné pour assurer la fonction de gestion de la configuration du logiciel.</p> <p>Il a pour tâche de :</p> <ul style="list-style-type: none"> ? définir les éléments à gérer, ? définir la version de référence à tout instant, ? connaître les relations entre éléments d'une configuration donnée (afin, par exemple de pouvoir répondre à la question : "quels sont les tests de validation qui correspondent à cette version du logiciel d'application ?"), ? définir les moments de mise en configuration (d'archivage) d'un élément de la configuration, ? définir et appliquer les modalités de livraison du produit 	Std	x
EX41	<p>Pour chaque version du logiciel, la gestion de configuration doit permettre :</p> <ul style="list-style-type: none"> ? d'identifier les documents avec leur référence et édition/révision ? de sauvegarder sur un support identifié avec une étiquette de marquage non décollable précisant le nom du logiciel d'application et sa version/révision: <ul style="list-style-type: none"> ✍ le produit (sources, données, procédures de génération), ✍ les jeux de tests de validation, ✍ les éventuels moyens d'essai spécifiques associés (simulateur, données, ...) ? de connaître les références : <ul style="list-style-type: none"> ✍ du support de sauvegarde (n° de bande par exemple), ✍ des faits techniques (anomalies, corrections, demandes de modification, ✍ des logiciels de base (système d'exploitation, émulateur, moniteur temps réel, bibliothèques, compilateurs, éditeurs de liens, ...), ✍ des matériels, ✍ des éventuels moyens d'essais ? de restaurer, si nécessaire, un logiciel archivé. 	Std	x

9.2.2 Cas des applications UNIX

Référence	Libellé	Hors R&T	Contexte R&T
Recom 2	Dans le cas de logiciels développés sur plate-forme UNIX, avec un besoin de portabilité sur plusieurs systèmes d'exploitation, l'utilisation de procédures standard de génération / installation est recommandée.	Std	x

Des solutions standard (à base des produits gratuits et libres d'utilisation : autoconf, automake, libtool) permettent de s'affranchir des spécificités des machines (nom du compilateur, chemins d'accès aux bibliothèques) lors de la génération, de paramétrer les répertoires d'installation des produits, d'obtenir des procédures simples pour lancer la génération et l'installation.

9.2.3 Document descriptif de la configuration

Référence	Libellé	Hors R&T	Contexte R&T
EX42	<p>Le fournisseur doit établir pour chaque version un dossier descriptif de la configuration (DDC), inclus dans la livraison de la version du produit.</p> <p>Celui-ci contient au minimum les fichiers suivants (cf. DR16 pour la description du contenu attendu de ces fichiers) :</p> <ul style="list-style-type: none"> ? ddc.txt.....identification de l'article ? fa-couv.txt.....liste des FA couvertes ? dm-couv.txt.....liste des DM couvertes ? etat-fa.txt.....liste des FA ouvertes ? ls-doc.txt.....liste des documents (CNES et fournisseur) associés au produit ? ls-ref.txt.....liste des fichiers constituant le produit 	Std	x

9.3 METHODES, OUTILS, REGLES

9.3.1 Principes

Référence	Libellé	Hors R&T	Contexte R&T
-----------	---------	----------	--------------

Référence	Libellé	Hors R&T	Contexte R&T
EX43	Les méthodes et éventuels outils support adaptés aux différentes phases du cycle de développement du logiciel sont choisis en fonction des compétences de l'équipe projet (plan de formation, expériences antérieures, ...) et de ses moyens financiers. L'utilisation de logiciels libres est fortement préconisée (en respectant les conditions de licence)	Std	x
EX44	Le choix des outils doit être justifié dans le plan d'application ou son équivalent.	Std	x
Recom 3	L'utilisation d'outils standards est préconisée, de manière homogène sur l'ensemble du développement.	Std	x

9.3.2 Règles de nomenclature

Référence	Libellé	Hors R&T	Contexte R&T
EX45	Des règles de nomenclature sont définies, en particulier pour les variables d'interface. Ces règles sont cohérentes avec les conventions d'usage pour le langage de codage choisi.	Std	x

9.3.3 Règles de codage

Référence	Libellé	Hors R&T	Contexte R&T
EX46	Les variables d'interface sont décrites sous forme de commentaires dans le code.	Std	x
EX47	Les règles d'auto-documentation intrinsèques au langage de programmation (exemple : commentaires exploitables par javadoc pour le langage Java) doivent être utilisées.	Std	x

Référence	Libellé	Hors R&T	Contexte R&T
EX48	? Chaque module de code comporte une entête commentaire contenant au moins les rubriques suivantes: <ul style="list-style-type: none"> ✍ nom du projet ✍ nom du module ✍ version ✍ auteur ✍ date ✍ rôle et but du module 	Std	x
	? Un seul point d'entrée, un seul point de sortie pour toute unité compilable.		
	? Toute partie de code doit être accessible (pas de code mort).		
	? Chaque ordre de saut conditionnel ou de rupture de séquence est commenté en expliquant l'objet du débranchement.		
	? Chaque fonction ou sous-programme doit préciser ses paramètres en entrée, en sortie ou en entrée/sortie, avec pour chacun : <ul style="list-style-type: none"> ✍ son nom, ✍ sa signification, ✍ l'unité utilisée en cas de grandeur physique, ✍ son degré de précision, ✍ son domaine de valeur 		

9.4 CONDITIONS PARTICULIERES DE DEVELOPPEMENT

9.4.1 Sûreté de fonctionnement et sécurité des logiciels

Référence	Libellé	Hors R&T	Contexte R&T
EX49	Une analyse fonctionnelle au niveau système doit être utilisée pour identifier les modules logiciels critiques, en tenant compte de l'interaction du logiciel avec son environnement.	Crit.	
Recom 4	Le fournisseur doit s'efforcer de concevoir les composants logiciels critiques aussi simplement que possible afin de faciliter l'analyse de la sûreté de fonctionnement et de la sécurité et les essais du logiciel.	Crit.	

9.4.2 Logiciels réutilisés

Référence	Libellé	Hors R&T	Contexte R&T
EX50	<p>Dans le cas où l'utilisation de logiciels existants (produits du marché, logiciels libres, autres produits) est envisagée, il faut tenir compte, dans le choix de ceux-ci, des éléments suivants :</p> <ul style="list-style-type: none"> ? l'évaluation du produit par rapport aux besoins, ? les conditions de recette et de garantie (démonstration de bon fonctionnement), et/ou de maintenance, ? les conditions de formation et d'utilisation, ? les contraintes de propriété industrielle et de licence (droit d'utilisation, de diffusion ou de modification). 	Std	x
EX51	<p>Pour les logiciels à adapter, il est indispensable, préalablement à toute modification :</p> <ul style="list-style-type: none"> ? d'évaluer le taux de modifications, ? de vérifier l'état de la documentation des logiciels existants afin de s'assurer de leur cohérence et complétude vis à vis du logiciel réutilisé, ? d'analyser l'impact des modifications sur la documentation et les tests nécessaires à la validation des modifications (test unitaire du constituant modifié, test fonctionnel mettant en jeu le constituant modifié, tests de non-régression de validation de l'ensemble du logiciel). 	Std	x
EX52	<p>A l'issue de la première phase du projet (phase de spécification initiale), la liste des logiciels que l'on envisage de réutiliser est présentée en précisant leur état et l'effort de développement (mise à niveau de la documentation, mise à niveau du code, tests nécessaires) associé en fonction des éléments justificatifs ci-dessus.</p>	Std	x
EX53	<p>Afin de faciliter la maintenance du logiciel, toute modification du code doit être réalisée de façon à conserver le squelette du constituant (pas de "verrue"). Si cette règle ne peut être satisfaite, on opte pour une reprogrammation complète du constituant en respectant les principes de la programmation structurée.</p>	Std	x
EX54	<p>La gestion en configuration des logiciels réutilisés doit être mise en place.</p>	Std	x
Recom 5	<p>La gestion en configuration autonome des composants logiciels réutilisés est recommandée. Ces composants logiciels réutilisés sont alors référencés dans le dossier descriptif de la configuration du produit développé.</p>	Std	x

9.4.3 Interface Homme – Machine

Référence	Libellé	Hors R&T	Contexte R&T
EX55	<p>Pour les logiciels disposant d'une interface homme/machine, une maquette de cette interface doit être établie pendant la phase de spécifications.</p> <p>Cette maquette doit présenter:</p> <ul style="list-style-type: none"> ? le contenu des écrans, ? la dynamique d'enchaînement des écrans, ? la mise en œuvre des différentes commandes, ? les aides en ligne éventuelles. 	Std	x
EX56	Les futurs utilisateurs doivent participer à la validation de la maquette de l'interface homme-machine, qui doit avoir lieu au cours de la phase de spécifications.	Std	
EX57	Les futurs utilisateurs sont impliqués dans la validation de la maquette. Dans le cas où la maquette est enrichie lors de cycles itératifs, les utilisateurs doivent participer à la validation à chaque pas du cycle.		x
EX58	Les utilisateurs doivent participer à la phase de recette du logiciel	Std	

9.5 DOCUMENTATION PRECONISEE DANS LE CAS D'UN DEVELOPPEMENT LOGICIEL STANDARD

Les documents de référence (DR) du chapitre 5 peuvent servir de base de travail à la rédaction de certains documents exigés du fournisseur.

Le tableau suivant présente en colonne les réunions formelles préconisées dans le cas d'un cycle de vie en V classique. Il convient d'adapter les instants de livraison des documents dans le cas d'un cycle de développement différent.

	Réunion de démarrage	RSL	Fin conception	BT Avant validation	CRE Après validation	Recette	A chaque version	A chaque réunion d'avancement
Plan d'application	P	D						
Document de spécification du logiciel		D						
Document de conception			D					
Document de validation		P		C	D			
Dossier descriptif de la configuration				C	J	J	J	
Manuel d'utilisation		P			C	D		
Cahier de recette			P			D		
Planning	P							J

Légende :

- P version préliminaire
- C version complète
- D version définitive
- J version mise à jour

Des précisions sur le contenu des documents (y compris les plans type) sont indiquées dans les annexes.

9.6 DOCUMENTATION PRECONISEE DANS LE CAS D'UN DEVELOPPEMENT LOGICIEL DE R&T

La rédaction d'un seul document est préconisée, dans le but d'alléger la gestion de la documentation dans ce contexte.

La documentation doit contenir les chapitres suivants :

- ? Contexte de développement
 - ✍ hypothèses prises pour le développement,
 - ✍ limitations des données d'entrée
- ? Spécifications
 - ✍ spécifications logicielles
- ? Informations destinées aux utilisateurs
 - ✍ Manuel d'utilisation (dans le cas d'une bibliothèque, exemples d'appels aux fonctions de la bibliothèque)
 - ✍ Manuel de référence (catalogue des fonctions disponibles)
- ? Informations nécessaires à la maintenance du produit développé
 - ✍ Liste des essais permettant de valider l'acceptation du produit
 - ✍ Documentation détaillée de conception (la préférence est donnée à l'auto-documentation dans les modules de code)
 - ✍ Historique des choix réalisés sur le développement (choix validés en réunion d'avancement et réunion technique avec le client)
 - ✍ Dossier descriptif de la configuration.

10 ANNEXE 1 : PLANS-TYPE

10.1 PLAN D'APPLICATION

10.1.1 Présentation

Il décrit les dispositions de gestion qui sont appliquées au projet en réponse aux exigences du présent document. Ces dispositions de gestion doivent être définies lors de réunion de travail entre le client et le fournisseur.

10.1.2 Plan-type du plan d'application

GLOSSAIRE

1. INTRODUCTION

1.1. DESCRIPTION SUCCINCTE DU PRODUIT

1.2. ELEMENTS CONCERNES PAR LE PLAN D'APPLICATION

2. DOCUMENTS APPLICABLES

3. DOCUMENTS DE REFERENCE

4. PLAN DE DEVELOPPEMENT

4.1. ORGANISATION DU PROJET

4.2. CYCLE DE DEVELOPPEMENT

4.2.1. Phase de spécification

4.2.2. Phase de conception

4.2.3. Phase de codage-tests unitaires

4.2.4. Phase de validation

4.2.5. Phase de recette

4.3. ORGANISATION DES REUNIONS

4.3.1. Réunions d'avancement

4.3.2. Points-clés et revues

4.4. MOYENS MATERIELS ET LOGICIELS

4.5. PLANNING

- 5. PLAN D'ASSURANCE PRODUIT**
- 5.1. SURETE DE FONCTIONNEMENT**
- 5.2. METHODES, OUTILS ET REGLES**
- 5.3. GESTION DE LA CONFIGURATION**
- 5.4. GESTION DE LA DOCUMENTATION**
- 5.5. GESTION DES MODIFICATIONS**
- 5.6. LOGICIELS REUTILISES**
- 5.7. INTERFACE HOMME-MACHINE**

10.2 DOCUMENT DE SPECIFICATION

10.2.1 Présentation

Il décrit :

- les fonctionnalités du logiciel,
- la manière dont le logiciel doit être opéré,
- les interfaces externes,
- les données manipulées,
- le flux de données,
- les éventuelles conditions d'activation,
- les éventuels points de reprise ou de sauvegarde de données à prévoir pour les cas dégradés,
- les contraintes (ressources,...),
- les conditions de réutilisation de produits tiers (COTS, logiciels libres, ...).

10.2.2 Plan-type du document de spécification

GLOSSAIRE

1. OBJET

2. DOCUMENTS APPLICABLES

3. DOCUMENTS DE REFERENCE

4. SPECIFICATIONS FONCTIONNELLES

4.1. PRESENTATION DE LA MISSION DU LOGICIEL

4.2. SPECIFICATIONS DE LA FONCTION <I>

4.3. SPECIFICATIONS DE LA FONCTION <J>

5. SPECIFICATIONS OPERATIONNELLES

5.1. DESCRIPTION DE L'ENVIRONNEMENT NECESSAIRE

5.1.1. Environnement matériel

5.1.2. Environnement logiciel

5.2. CONDITIONS D'INITIALISATION / ARRET / REPRISE

5.3. CONDITIONS D'ACTIVATION DES FONCTIONNALITES

6. SPECIFICATIONS DES PERFORMANCES

7. SPECIFICATIONS DES INTERFACES EXTERNES

7.1. DONNEES EN ENTREE

7.2. DONNEES EN SORTIE

8. SPECIFICATIONS DES INTERFACES INTERNES

8.1. DESCRIPTION DE LA DONNEE <I>

8.2. DESCRIPTION DE LA DONNEE <J>

9. EVALUATION DES LOGICIELS REUTILISES

9.1. LOGICIEL <A>

9.1.1. Présentation du logiciel

9.1.2. Intérêt de la réutilisation

9.1.3. Modifications envisagées sur le logiciel

Sur le produit, sur sa documentation et sur ses tests

10. CONFORMITE A LA SPECIFICATION DE BESOINS

Sous la forme d'une matrice de couverture si la spécification de besoins est structurée en exigences ou en paragraphes.

10.3 DOCUMENT DE CONCEPTION

10.3.1 Présentation

Il décrit la solution retenue pour répondre aux spécifications du logiciel:

- l'architecture du logiciel avec la décomposition en constituants (graphes d'appel, flux de données),
- les éventuelles synchronisations entre constituants,
- la stratégie de traitement des erreurs et des exceptions.

La description de chaque constituant apparaît en commentaires dans le code. Elle comporte :

- son rôle d'un point de vue fonctionnel,
- ses principales interfaces (fichiers, paramètres, messages, ...),
- ses "liens" avec d'autres constituants (liens d'appel, de dépendance, de service, d'héritages, de généricité, ...),
- ses éventuelles conditions d'activation,
- son pseudo-code.

10.3.2 Plan-type du document de conception

GLOSSAIRE

1. OBJET

2. DOCUMENTS APPLICABLES

3. DOCUMENTS DE REFERENCE

4. DEMARCHE DE CONCEPTION

4.1. PRESENTATION DE LA DEMARCHE DE CONCEPTION

4.2. CONTRAINTES

4.3. STRATEGIE DE GESTION DES ERREURS

5. ARCHITECTURE DU LOGICIEL

5.1. PRESENTATION DES PRINCIPES DE L'ARCHITECTURE

5.2. ORGANISATION DE LA CONCEPTION

Schémas de conception

6. CONCEPTION DES DONNEES

6.1. ARBORESCENCE DES DONNEES SUR DISQUE

6.2. FLUX DES DONNEES

10.4 DOCUMENT DE VALIDATION

10.4.1 Présentation

En fin de phase de spécifications du logiciel, une première version de ce document décrit :

- l'organisation, la planification, les enchaînements des tests,
- les ressources nécessaires, en particulier les moyens d'essais en précisant les limites de leur représentativité par rapport à l'environnement réel:
 - limites des domaines physiques,
 - limites de performances,
 - limites en terme de fonctionnalités,
 - limites en terme de commandabilité et d'observabilité,
- le plan de validation (description des éléments de logiciels à tester d'un point de vue fonctionnel, liste des tests en terme d'objectifs et de moyens nécessaires).

Avant le début de la phase de validation du logiciel, ce document est complété afin de décrire, pour chaque test de validation, les procédures de test. Ces procédures comportent la description de la mise en oeuvre des tests, des actions à réaliser avant, pendant ou après le test, des moyens à mettre en place, et des résultats attendus.

En fin de phase de validation du logiciel, le document final doit être complété par les compte-rendus de tests qui décriront les résultats obtenus avec les références des éventuelles anomalies détectées.

10.4.2 Plan-type du document de validation

GLOSSAIRE

- 1. INTRODUCTION**
- 2. DOCUMENTS APPLICABLES**
- 3. DOCUMENTS DE REFERENCE**
- 4. RAPPEL DES FONCTIONNALITES DU PRODUIT**
- 5. ARCHITECTURE DU PRODUIT**
- 6. ORGANISATION DE LA VALIDATION**
 - 6.1. ENVIRONNEMENT**

6.1.1. CONFIGURATION ET SITE

6.1.2. MOYENS D'ESSAIS

6.2. PLANNING

6.3. LOGIQUE D'ENCHAINEMENT DES ESSAIS

7. PLAN DE VALIDATION

7.1. LOGICIELS A TESTER

7.2. INSTALLATION - PARAMETRAGE

7.3. LES TESTS FONCTIONNELS

7.4. LES CONTRAINTES

7.4.1. CONTRAINTES D'ENVIRONNEMENT

7.4.2. PERFORMANCES

8. DESCRIPTION DES TESTS

Projet XXXXX		
Test N° : xxxxxxxx		
Essai : XXXXX	Date : 21/04/2003	Version du test : 1.0
Type : Fonctionnel [<input type="checkbox"/>]	Dégradé [<input type="checkbox"/>]	Performances [<input type="checkbox"/>]
OBJECTIF DU TEST :		
TESTS PREALABLES :		
DONNEES DE TEST :		
RESULTATS ATTENDUS :		
PROCEDURE DE MISE EN OEUVRE :		
RESULTATS OBTENUS :		
Référence des fiches d'anomalies :		
Commentaires :		

10.5 MANUEL D'UTILISATION

10.5.1 Présentation

La fourniture de ce document dépend du type de logiciel. Elle est indispensable pour les logiciels disposant d'une interface homme/machine.

Ce document décrit :

- les différents types d'utilisateurs possibles,
- les moyens d'interaction (clavier, souris, touches de fonction, ascenseurs, boutons, ...), leur signification et leurs principes d'utilisation,
- les menus et grilles d'écran : type, signification des différents champs, contrôles effectués et actions associées,
- les messages émis par le logiciel avec indication de la fonction émettrice et les actions correctives associées,
- l'enchaînement des écrans,
- la forme des éditions papier.

L'organisation et le contenu de ce document sont à soumettre à l'approbation des futurs utilisateurs.

Ce document est initialisé dès la phase de spécifications du logiciel et est complété au cours du développement afin d'être livré en version complète en fin de validation du logiciel.

10.5.2 Plan-type du manuel d'utilisation

GLOSSAIRE

1. INTRODUCTION

1.1. OBJECTIFS

1.2. UTILISATION DU DOCUMENT

1.3. CONVENTIONS D'ECRITURE

2. CONCEPTS TECHNIQUES GENERAUX

2.1. PRESENTATION GENERALE DU PRODUIT

2.2. PRESENTATION DES FONCTIONNALITES ET DES INTERVENANTS

2.3. SPECIFICITES TECHNIQUES

2.4. LA BASE DE DONNEES

3. ENVIRONNEMENT DU SYSTEME

3.1. ORGANISATION

3.2. RESSOURCES

3.3. DROIT D'ACCES

4. DEMARRAGE DE L'APPLICATION

5. DESCRIPTION DE L'IHM

6. UTILITAIRES GENERAUX

6.1. ADMINISTRATION DES UTILISATEURS

6.2. ADMINISTRATION DU SYSTEME

6.3. LE JOURNAL DE BORD

7. DETAIL DES FONCTIONS

7.1. < NOM DE LA FONCTION 1 >

7.1.1. Description opérationnelle

7.1.2. Description de l'environnement

7.1.3. Démarrage de la fonction

7.1.4. Dynamique de la fonction

7.1.5. Les services

Pour chaque service, on donne sa description, ses conditions d'utilisation, sa mise en œuvre, des exemples, les messages d'erreur.

7.1.6. Messages d'erreurs et reprises

7.1.7. Cas dégradés

7.2. < NOM DE LA FONCTION 2 >

7.3. ...

8. INDEX GENERAL

10.6 CAHIER DE RECETTE

10.6.1 Présentation

Les essais de recette décrits dans ce document sont un extrait de la liste des tests du document de validation du logiciel. Le cahier de recette doit être approuvé par le CNES avant le début de la recette.

10.6.2 Plan-type du cahier de recette

GLOSSAIRE

1. INTRODUCTION

2. RAPPEL DES FONCTIONNALITES DU PRODUIT

3. ARCHITECTURE DU PRODUIT

4. COUVERTURE DE LA RECETTE

4.1. FOURNITURES A RECETTER

4.1.1. LES LOGICIELS

Liste des logiciels spécifiques avec leur référence et leur version

Liste des logiciels standards (ou sur étagère) avec leur référence et leur version

4.1.2. LES DOCUMENTS

Liste des documents avec leur référence et leur version

4.1.3. LES MATERIELS

Liste des matériels avec leur référence et leur version

4.2. CONTEXTE DE LA RECETTE

Préciser par exemple en quoi la recette est partielle (fonctions, tests, ...)

5. LES ESSAIS DE RECETTE

5.1. INSTALLATION - PARAMETRAGE

5.2. LES TESTS FONCTIONNELS

5.3. LES CONTRAINTES

5.3.1. CONTRAINTES D'ENVIRONNEMENT

5.3.2. PERFORMANCES

6. LOGIQUE D'ENCHAÎNEMENT DES ESSAIS

7. ORGANISATION DE LA RECETTE

7.1. ENVIRONNEMENT

7.1.1. CONFIGURATION ET SITE

7.1.2. MOYENS D'ESSAIS

7.2. PLANNING

7.3. INTERVENANTS

7.4. RESULTATS DE LA RECETTE

8. ANNEXE 1 - MATRICE DE COUVERTURE DES EXIGENCES

9. ANNEXE 2 - FAITS TECHNIQUES INCLUS

10. ANNEXE 3 - FAITS TECHNIQUES NON COUVERTS

11. FICHE DE TEST TYPE

Projet XXXXX		
Test de Recette N° : xxxxxxxx		
Essai : XXXXX	Date :	Version du test : 1.0
Type : Fonctionnel [<input type="checkbox"/>]	Dégradé [<input type="checkbox"/>]	Performances [<input type="checkbox"/>]
OBJECTIF DU TEST :		
TESTS PREALABLES:		
DONNES DE TEST :		
RESULTATS ATTENDUS :		
PROCEDURE DE MISE EN OEUVRE :		
RESULTATS :	TEST	ACCEPTÉ [<input type="checkbox"/>] REFUSE [<input type="checkbox"/>]
Référence des fiches d'anomalies :		
Commentaires :		

11 ANNEXE 2 : INDEX DES REGLES

11.1 CAS D'UN DEVELOPPEMENT LOGICIEL STANDARD

Code Règle	Rubrique	Thème	Page
EX1	Principes fondamentaux de la méthode	Règles pré-établies	11
EX2	Principes fondamentaux de la méthode	Plan d'application	11
EX3	Organisation et assurance produit	Organisation et responsabilités	11
EX4	Organisation et assurance produit	Formations	11
EX5	Organisation et assurance produit	Dispositions d' Assurance Produit	11
EX6	Organisation et assurance produit	Assurance Produit minimum	11
EX7	Organisation et assurance produit	Compléments d' Assurance Produit	11
EX8	Organisation et assurance produit	Identification des risques techniques	11
EX9	Organisation et assurance produit	Analyse d'impact des risques système	11
EX10	Organisation et assurance produit	Sous-traitants	11
EX11	Cycle de développement	Définition du cycle de développement	12
EX12	Cycle de développement	Analyse de besoins	12
EX14	Cycle de développement	Phases du cycle de développement	12
EX16	Cycle de développement	Jalon de fin de phase	12
EX18	Spécification du logiciel	Activités de spécifications minimum	13
EX19	Spécification du logiciel	Activités de spécifications complémentaires	13
EX20	Spécification du logiciel	Document de spécifications du logiciel	13
EX21	Spécification du logiciel	Maquette IHM	13
EX23	Spécification du logiciel	Revue de spécifications du logiciel (RSL)	13
EX24	Conception	Activités de conception minimum	14
EX26	Conception	Document de conception du logiciel	14
EX27	Conception	Conception des fonctions réutilisées	14
EX28	Codage - Tests unitaires	Activités de codage et de tests unitaires	14
Recom1	Codage - Tests unitaires	Langage de programmation	14
EX29	Codage - Tests unitaires	Respect des règles de codage	14
EX30	Validation	Activités de validation	15
EX31	Validation	Couverture des essais de validation	15
EX34	Validation	Stabilité version logiciel	15
EX35	Validation	BT - CRE	16
EX37	Recette	Phase de recette	17
EX39	Gestion de la configuration - Principes	Identification des éléments	18

Code Règle	Rubrique	Thème	Page
EX40	Gestion de la configuration - Principes	Responsable de gestion de configuration	18
EX41	Gestion de la configuration - Principes	Activités de gestion de configuration	18
Recom2	Gestion de la configuration - UNIX	Procédures standard de génération / installation	19
EX42	Gestion de la configuration - DDC	Dossier descriptif de la configuration	19
EX43	Méthodes, outils, règles - Principes	Choix des méthodes et outils	19
EX44	Méthodes, outils, règles - Principes	Justification du choix des outils	19
Recom3	Méthodes, outils, règles - Principes	Utilisation d'outils standards	20
EX45	Méthodes, outils, règles - Nomenclature	Règles de nomenclature	20
EX46	Méthodes, outils, règles - Règles de codage	Variables d'interface	20
EX47	Méthodes, outils, règles - Règles de codage	Auto-documentation	20
EX48	Méthodes, outils, règles - Règles de codage	Règles de codage essentielles	20
EX50	Logiciels réutilisés	Critères de choix pour l'utilisation de logiciels existants	22
EX51	Logiciels réutilisés	Evaluation des logiciels à adapter	22
EX52	Logiciels réutilisés	Etat des logiciels réutilisés	22
EX53	Logiciels réutilisés	Règles de modification du code	22
EX54	Logiciels réutilisés	Gestion en configuration des logiciels réutilisés	22
Recom5	Logiciels réutilisés	Gestion en configuration autonome des composants logiciels réutilisés	22
EX55	Interface Homme – Machine	Maquette de l'IHM	23
EX56	Interface Homme – Machine	Validation de la maquette IHM	23
EX58	Interface Homme – Machine	Participation des utilisateurs à la recette	23

11.2 CAS D'UN DEVELOPPEMENT LOGICIEL CRITIQUE

Code Règle	Rubrique	Thème	Page
EX1	Principes fondamentaux de la méthode	Règles pré-établies	11
EX2	Principes fondamentaux de la méthode	Plan d'application	11
EX3	Organisation et assurance produit	Organisation et responsabilités	11
EX4	Organisation et assurance produit	Formations	11
EX5	Organisation et assurance produit	Dispositions d' Assurance Produit	11
EX6	Organisation et assurance produit	Assurance Produit minimum	11
EX7	Organisation et assurance produit	Compléments d' Assurance Produit	11
EX8	Organisation et assurance produit	Identification des risques techniques	11
EX9	Organisation et assurance produit	Analyse d'impact des risques système	11
EX10	Organisation et assurance produit	Sous-traitants	11
EX11	Cycle de développement	Définition du cycle de développement	12
EX12	Cycle de développement	Analyse de besoins	12
EX14	Cycle de développement	Phases du cycle de développement	12
EX16	Cycle de développement	Jalon de fin de phase	12
EX17	Cycle de développement	Logiciel intégré dans un équipement	13
EX18	Spécification du logiciel	Activités de spécifications minimum	13
EX19	Spécification du logiciel	Activités de spécifications complémentaires	13
EX20	Spécification du logiciel	Document de spécifications du logiciel	13
EX21	Spécification du logiciel	Maquette IHM	13
EX22	Spécification du logiciel	Déterminer les parties critiques	13
EX23	Spécification du logiciel	Revue de spécifications du logiciel (RSL)	13
EX24	Conception	Activités de conception minimum	14
EX25	Conception	Activités de conception complémentaires	14
EX26	Conception	Document de conception du logiciel	14
EX27	Conception	Conception des fonctions réutilisées	14
EX28	Codage - Tests unitaires	Activités de codage et de tests unitaires	14
Recom1	Codage - Tests unitaires	Langage de programmation	14
EX29	Codage - Tests unitaires	Respect des règles de codage	14
EX30	Validation	Activités de validation	15
EX31	Validation	Couverture des essais de validation	15
EX33	Validation	Logiciel intégré dans un équipement	15
EX34	Validation	Stabilité version logiciel	15

Code Règle	Rubrique	Thème	Page
EX35	Validation	BT - CRE	16
EX37	Recette	Phase de recette	17
EX38	Recette	Logiciel intégré dans un équipement	17
EX39	Gestion de la configuration - Principes	Identification des éléments	18
EX40	Gestion de la configuration - Principes	Responsable de gestion de configuration	18
EX41	Gestion de la configuration - Principes	Activités de gestion de configuration	18
Recom2	Gestion de la configuration - UNIX	Procédures standard de génération / installation	19
EX42	Gestion de la configuration - DDC	Dossier descriptif de la configuration	19
EX43	Méthodes, outils, règles - Principes	Choix des méthodes et outils	19
EX44	Méthodes, outils, règles - Principes	Justification du choix des outils	19
Recom3	Méthodes, outils, règles - Principes	Utilisation d'outils standards	20
EX45	Méthodes, outils, règles - Nomenclature	Règles de nomenclature	20
EX46	Méthodes, outils, règles - Règles de codage	Variables d'interface	20
EX47	Méthodes, outils, règles - Règles de codage	Auto-documentation	20
EX48	Méthodes, outils, règles - Règles de codage	Règles de codage essentielles	20
EX49	Sûreté de fonctionnement	Analyse fonctionnelle	21
Recom4	Sûreté de fonctionnement	Conception composants logiciels critiques	21
EX50	Logiciels réutilisés	Critères de choix pour l'utilisation de logiciels existants	22
EX51	Logiciels réutilisés	Evaluation des logiciels à adapter	22
EX52	Logiciels réutilisés	Etat des logiciels réutilisés	22
EX53	Logiciels réutilisés	Règles de modification du code	22
EX54	Logiciels réutilisés	Gestion en configuration des logiciels réutilisés	22
Recom5	Logiciels réutilisés	Gestion en configuration autonome des composants logiciels réutilisés	22
EX55	Interface Homme – Machine	Maquette de l'IHM	23
EX56	Interface Homme – Machine	Validation de la maquette IHM	23
EX58	Interface Homme – Machine	Participation des utilisateurs à la recette	23

11.3 CAS D'UN DEVELOPPEMENT LOGICIEL R&T

Code Règle	Rubrique	Thème	Page
EX1	Principes fondamentaux de la méthode	Règles pré-établies	11
EX3	Organisation et assurance produit	Organisation et responsabilités	11
EX4	Organisation et assurance produit	Formations	11
EX5	Organisation et assurance produit	Dispositions d'Assurance Produit	11
EX6	Organisation et assurance produit	Assurance Produit minimum	11
EX8	Organisation et assurance produit	Identification des risques techniques	11
EX9	Organisation et assurance produit	Analyse d'impact des risques système	11
EX10	Organisation et assurance produit	Sous-traitants	11
EX11	Cycle de développement	Définition du cycle de développement	12
EX13	Cycle de développement	Analyse de besoins	12
EX15	Cycle de développement	Critères d'itération	12
EX16	Cycle de développement	Jalon de fin de phase	12
EX18	Spécification du logiciel	Activités de spécifications minimum	13
EX21	Spécification du logiciel	Maquette IHM	13
EX24	Conception	Activités de conception minimum	14
EX27	Conception	Conception des fonctions réutilisées	14
EX28	Codage - Tests unitaires	Activités de codage et de tests unitaires	14
Recom1	Codage - Tests unitaires	Langage de programmation	14
EX29	Codage - Tests unitaires	Respect des règles de codage	14
EX30	Validation	Activités de validation	15
EX32	Validation	Validation en développement itératif	15
EX34	Validation	Stabilité version logiciel	15
EX36	Validation	Réunions de début et fin de validation	17
EX37	Recette	Phase de recette	17
EX39	Gestion de la configuration - Principes	Identification des éléments	18
EX40	Gestion de la configuration - Principes	Responsable de gestion de configuration	18
EX41	Gestion de la configuration - Principes	Activités de gestion de configuration	18
Recom2	Gestion de la configuration - UNIX	Procédures standard de génération / installation	19
EX42	Gestion de la configuration - DDC	Dossier descriptif de la configuration	19
EX43	Méthodes, outils, règles - Principes	Choix des méthodes et outils	19
EX44	Méthodes, outils, règles - Principes	Justification du choix des outils	19

Code Règle	Rubrique	Thème	Page
Recom3	Méthodes, outils, règles - Principes	Utilisation d'outils standards	20
EX45	Méthodes, outils, règles - Nomenclature	Règles de nomenclature	20
EX46	Méthodes, outils, règles - Règles de codage	Variables d'interface	20
EX47	Méthodes, outils, règles - Règles de codage	Auto-documentation	20
EX48	Méthodes, outils, règles - Règles de codage	Règles de codage essentielles	20
EX50	Logiciels réutilisés	Critères de choix pour l'utilisation de logiciels existants	22
EX51	Logiciels réutilisés	Evaluation des logiciels à adapter	22
EX52	Logiciels réutilisés	Etat des logiciels réutilisés	22
EX53	Logiciels réutilisés	Règles de modification du code	22
EX54	Logiciels réutilisés	Gestion en configuration des logiciels réutilisés	22
Recom5	Logiciels réutilisés	Gestion en configuration autonome des composants logiciels réutilisés	22
EX55	Interface Homme – Machine	Maquette de l'IHM	23
EX57	Interface Homme – Machine	Validation de la maquette IHM	23



REFERENTIEL NORMATIF REALISE PAR :
Centre National d'Études Spatiales
Inspection Générale Direction de la Fonction Qualité
18 Avenue Edouard Belin
31401 TOULOUSE CEDEX 9
Tél. : 05 61 27 31 31 - Fax : 05 61 28 28 49

CENTRE NATIONAL D'ÉTUDES SPATIALES

Siège social : 2 pl. Maurice Quentin 75039 Paris cedex 01 / Tel. (33) 01 44 76 75 00 / Fax : 01 44 46 76 76
RCS Paris B 775 665 912 / Siret : 775 665 912 00082 / Code APE 731Z