SAINT-MALO
11 au 13 octobre 2016

Tutoriel A2:

Sûreté de fonctionnement des systèmes programmés

P. Kahn
KSdF-Conseil
C. Triolaire
Affinity Software

MAÎTRISER LES RISQUES DANS UN MONDE EN MOUVEMENT

Journée des tutoriels – 10/10/16

IMdR Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Plan du tutoriel

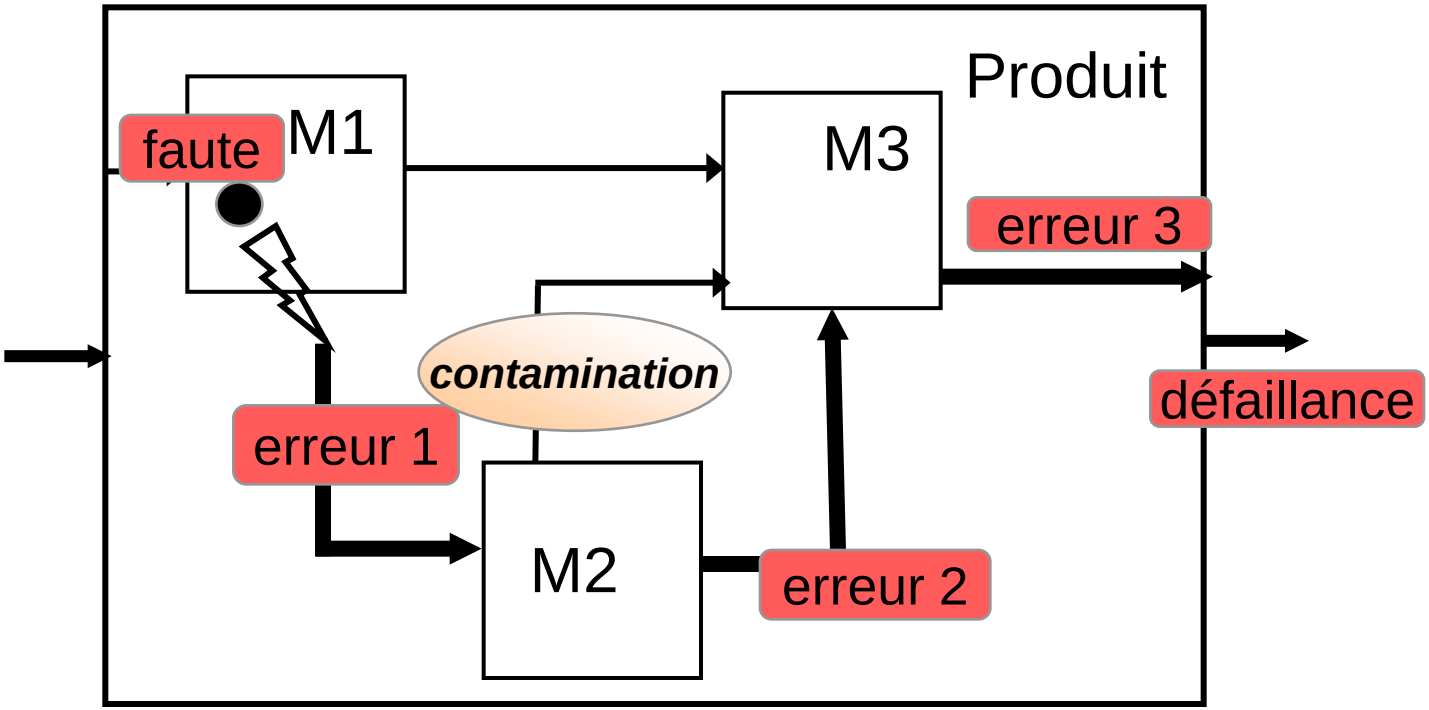
- Spécificités des logiciels
- Cycle de vie systèmes programmés et Sûreté de Fonctionnement
- Application de la SdF aux systèmes programmés
  - Techniques d'analyse
  - Architectures pour la Sûreté de Fonctionnement
  - Démarche de conception et de codage
  - Moyens de vérification
- Référentiels normatifs sécurité fonctionnelle des systèmes programmés:
  - Principes de normalisation
  - Norme générique CEI 61508 Ed.2
  - Déclinaisons pour le ferroviaire, l'automobile, …
  - Point de vue de l'aéronautique
- Conclusion

Spécificités des logiciels

- Caractéristiques du logiciel
  - Non susceptibilité aux usures et aux détériorations physiques
  - Susceptibilité aux erreurs de conception et de réalisation
  - Difficulté à estimer le nombre d'erreurs résiduelles
  - Difficulté à identifier les conditions de défaillance
  - Difficulté à déterminer la probabilité de défaillance
- Pas d’approche quantitative de la SdF
- Approche qualitative indirecte basée sur la démonstration de la qualité du développement

Principe de défaillance d'un logiciel

- Faute
- Erreur
- Défaillance



- Une faute de codage provoque une erreur dite latente
- Quand ce code est exécuté, l'erreur devient active
- Quand le comportement observé est différent du comportement spécifié, on constate une défaillance

Sécurité vs Sécurité

- Sécurité - innocuité (Safety) :
  - Aptitude d’un logiciel à éviter de faire apparaître, dans des conditions données, des événements critiques ou catastrophiques
- Sécurité - confidentialité (Security)
  - Aptitude d'un produit à protéger ses données et ses traitements contre des accès ou des modifications non autorisées
  - Elle est liée à la notion de malveillance

Sécurité – confidentialité (Security)

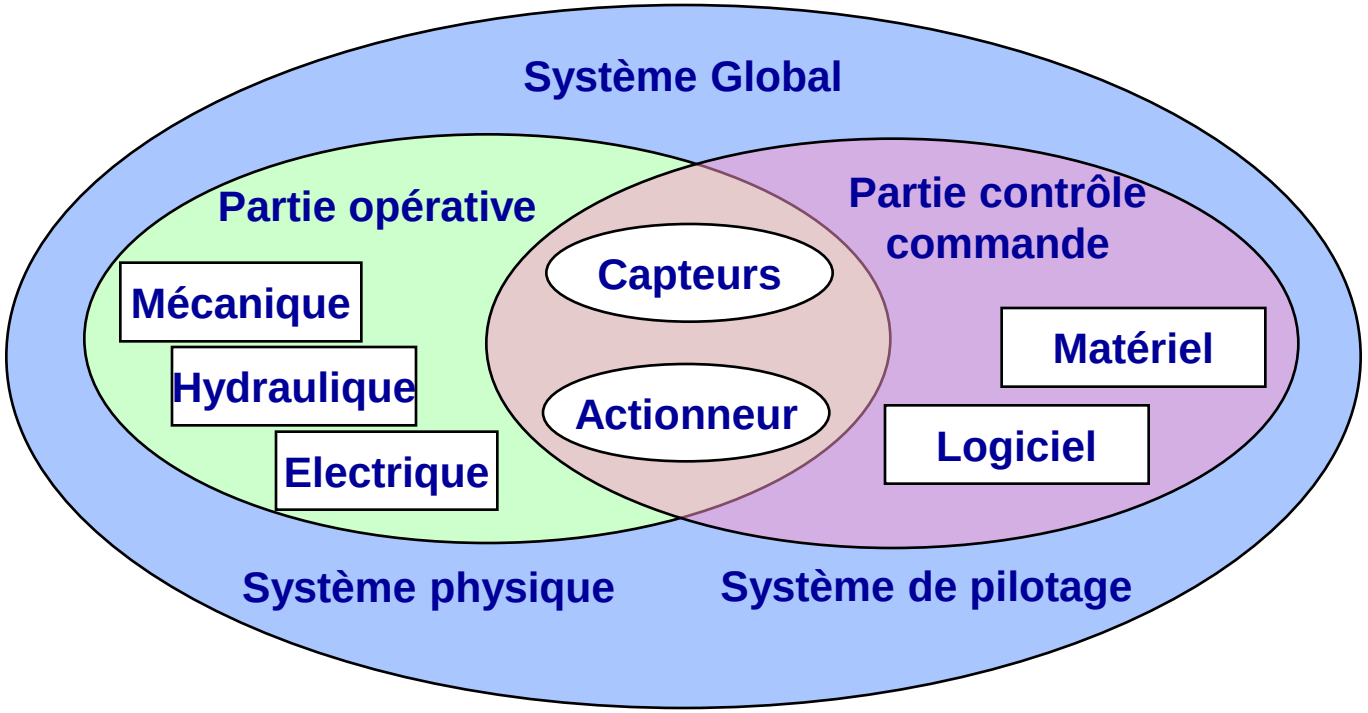
- Lié à la malveillance
- Démarches spécifiques :
  - Critères communs / Orange Book
  - Norme support : ISO 2700x
  - Analyse des vulnérabilités : Méthodes spécifiques : MARION, MELISSA, EBIOS, …
  - Pour plus d'informations : voir le site du clusif : www.clusif.asso.fr

Pour le logiciel, la maintenabilité c'est quoi ?

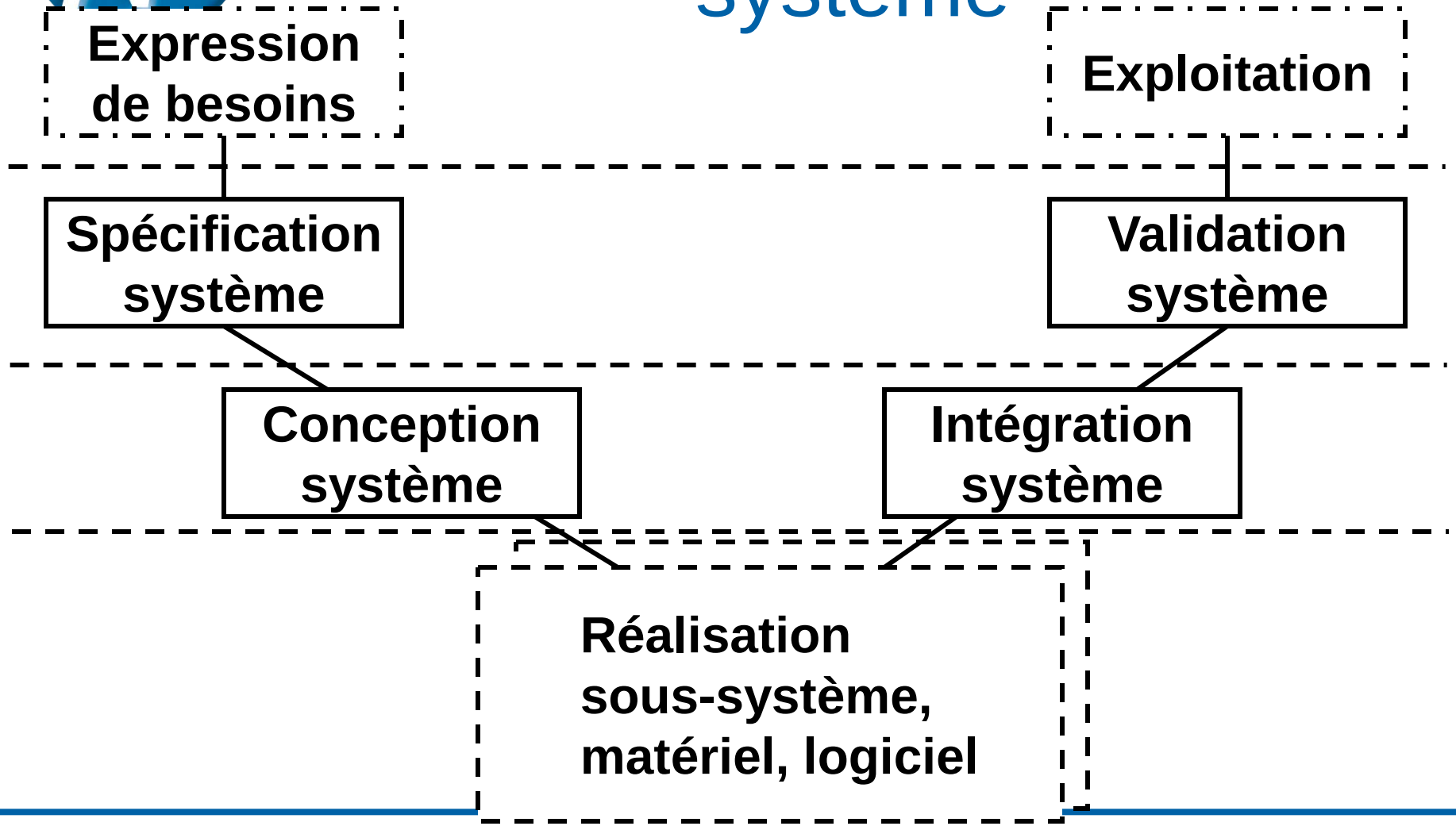
- Maintenabilité
  - Aptitude d'un système à être maintenu ou rétabli dans un état dans lequel il peut accomplir une fonction requise, lorsque l'entretien (maintenance) est accompli dans des conditions données, avec des procédures et des moyens prescrits (préventifs ou correctifs).
  - Pour les logiciels 2 volets à considérer :
    - Aptitude à remettre en état de fonctionnement
    - Aptitude du logiciel à être modifié (correction d'anomalies, évolutions). Elle est donc liée à la qualité du développement

Notion de système embarqué

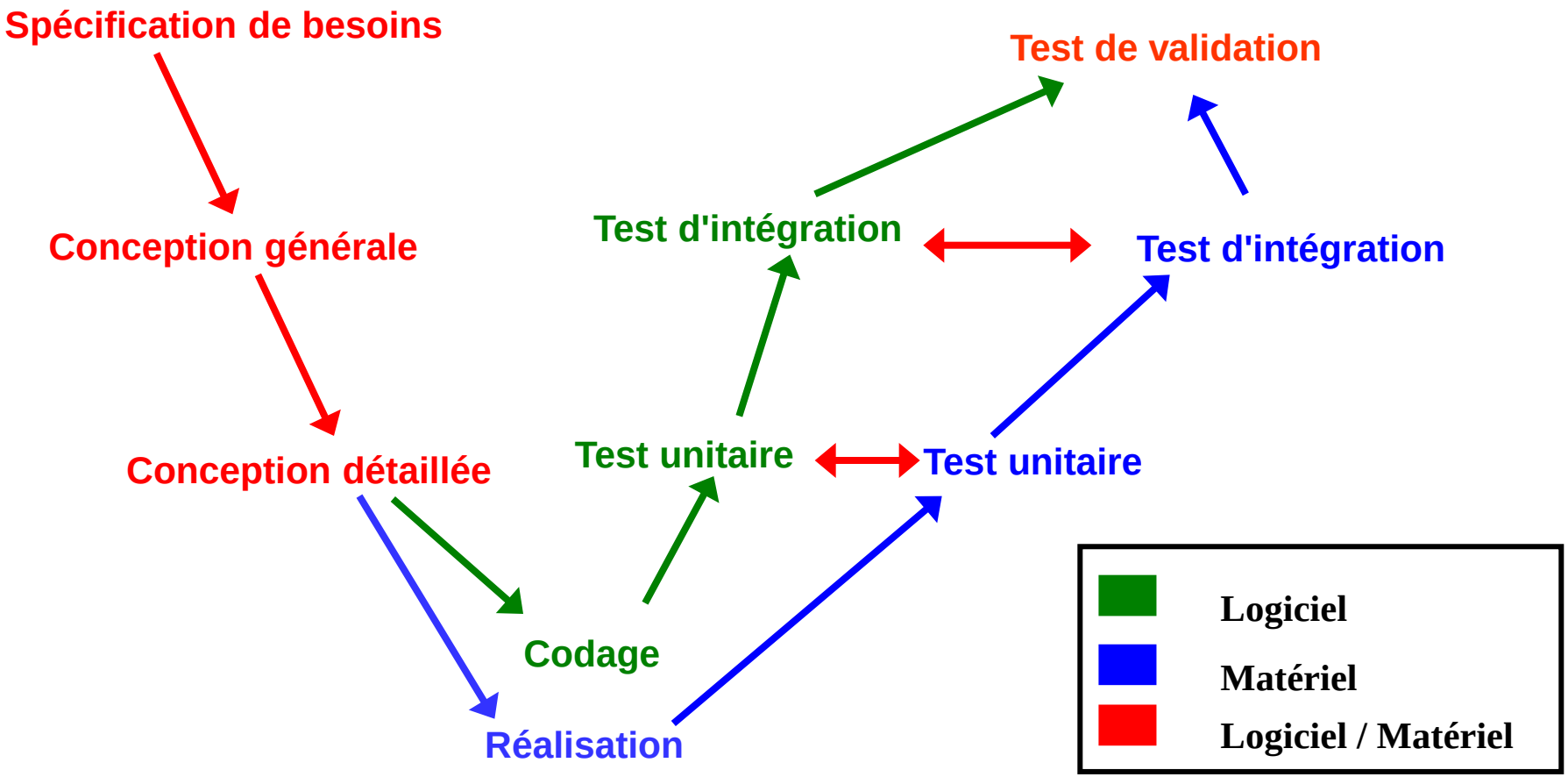
- Vision globale du système embarqué



Cycle de vie générique système



Co-conception matériel - logiciel



Activités SdF et cycles de vie

- Pas réellement de cycle de vie logiciel dédié SdF
- Renforcement des activités de chaque étape pour prendre en compte les exigences de SdF
- Pour chaque étape :
  - activités de construction de la SdF
  - activités de vérification de la SdF

APPLICATION DE LA SDF AUX SYSTÈMES PROGRAMMÉS

➔ Techniques d’analyse

– Architectures pour la Sûreté de Fonctionnement

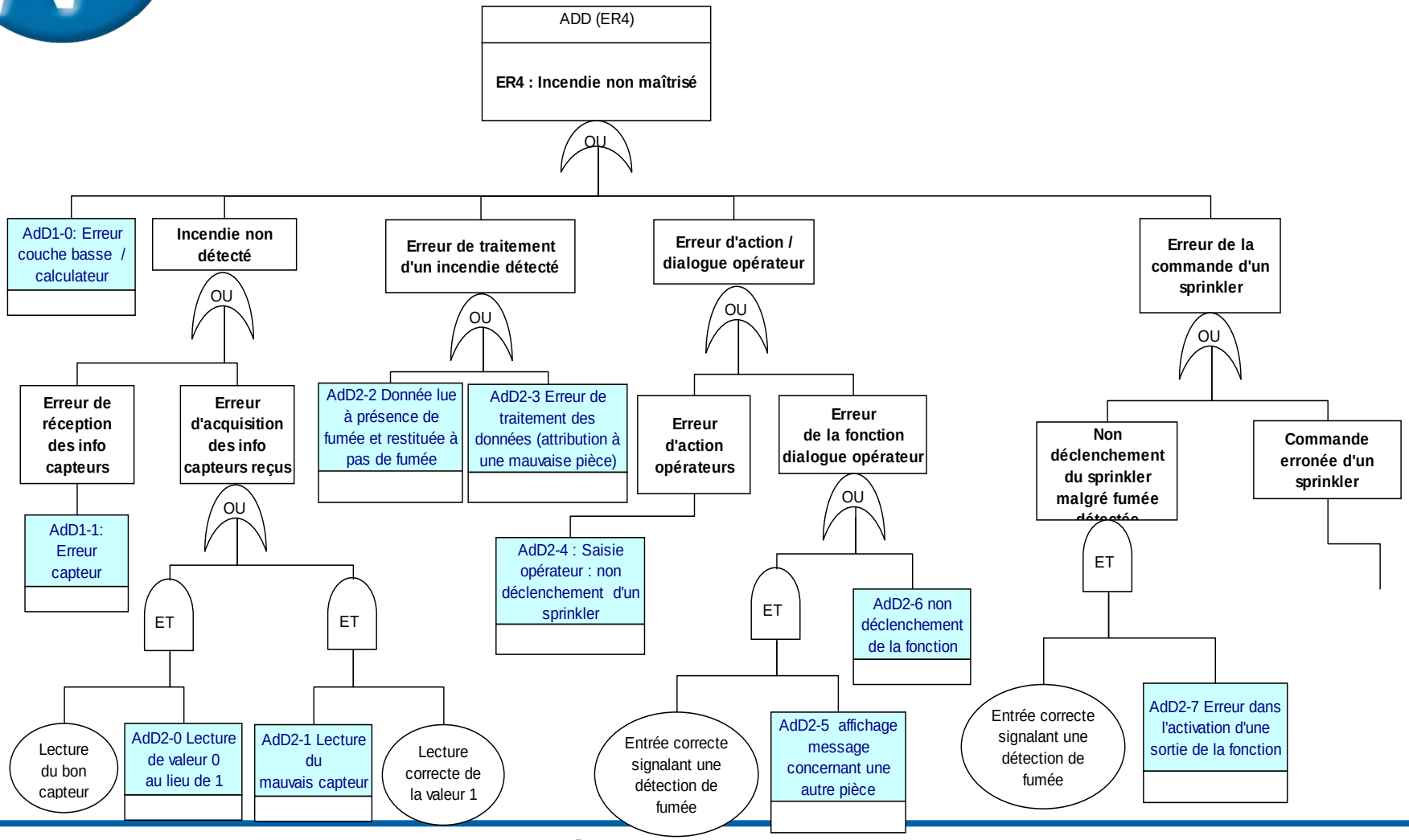
– Démarche de conception et de codage

– Moyens de vérification

Analyse de SdF

- Analyse de SdF :
  - Analyse déductive : type arbre de défaillances
  - Analyse inductive : type AMDE (Analyse des Modes de Défaillance et de leurs Effets)
  - Analyse comportementale : type Réseau de Pétri
- Menées en parallèle avec activités de spécification et de conception, avec en entrée :
  - Evénements redoutés pour le système et/ou le logiciel
  - Description du logiciel (spécification et architecture)
- Objectif : identifier des actions en réduction de risque et classer les logiciels en criticité

Exemple d'arbres de fautes



Exemple d'AMDE Logiciel

#	designation de l'équipement / fonction	fonction / sous-fonction	mode de défaillance	cause défaillance	Effet de Défaillance					Moyens de Détection de la défaillance	Moyen de réduction de risque	Effet résultant			actions	status
					effet local (pire cas)	effet final (pire cas)	Gravité	Prob.	Criticité			Gravité	Prob.	Criticité		
1	Surveillance Incendie	Toutes les fonctions	Perte des fonctions / du calculateur	non respect des règles de programmation	Perte de surveillance de la ou des fonctions concernées	ER4 Incendie non maîtrisé **et/ou** ER3 : Incendie signalé à tort avec inondation	3	3	9	Surveillance temps réel des fonctions actives		3	2	6		
2	Surveillance Incendie	Toutes les fonctions	Déclenchement intempestif	*voir chaque fonction*	*voir chaque fonction*	*voir chaque fonction*										
3	Surveillance Incendie	Toutes les fonctions	Fourniture de données erronées	*voir chaque fonction*	*voir chaque fonction*	*voir chaque fonction*										
4	Surveillance Incendie	Toutes les fonctions	Arrêt de la fonction	*voir cas perte ci-dessus (fonctions continues)*	*voir cas perte ci-dessus (fonctions continues)*	*voir cas perte ci-dessus (fonctions continues)*										
5	Surveillance Incendie	Acquisition capteur	Aucune acquisition des données	Non lancement de la fonction	Aucune pièce surveillée	ER4 Incendie non maitrisé	3	2	6	Surveillance temps réel des fonctions actives	mécanisme de synchro des tâches (présence flag)	3	1	3		
6	Surveillance Incendie	Acquisition capteur	Non acquisition aléatoire de l'info d'un capteur	Erreur dans l'algorithme de scrutation des capteurs	Surveillance partielle : non faite pour un capteur aléatoirement	ER4 Incendie non maitrisé	3	2	6	Compteur systématique du nombre de capteurs interrogés	programmation défensive	3	1	3		
7	Surveillance Incendie	Acquisition capteur	Non acquisition systématique de l'info d'un capteur	Erreur dans l'algorithme de scrutation des capteurs	Capteur non surveillé	ER4 Incendie non maitrisé	3	3	9	Compteur systématique du nombre de capteurs interrogés	Validation de l'algorithme	3	1	3		
8	Surveillance Incendie	Acquisition capteur	Données erronées à fumée non détectée	Info capteur erronée	Erreur d'information fournie par le capteur	ER4 Incendie non maitrisé	3	3	9	Aucun	Redondance capteur ou changement de gamme	3	2	6		
9	Surveillance Incendie	Acquisition capteur	Données erronées à fumée détectée	Info capteur erronée	Erreur d'information fournie par le capteur	ER3 : Incendie signalé à tort avec inondation	2	3	6	Aucun	Redondance capteur ou changement de gamme	2	2	4		

Actions en réduction de risque à mettre en œuvre

- Lors des activités de spécification :
  - Modes dégradés
  - Invariants de sécurité
  - Analyse de probabilité (modélisation, simulation)
  - Contraintes sur architecture (code correcteur, diversification, …)
  - Identification des tests spécifiques

Actions en réduction de risque à mettre en œuvre (suite)

- Lors des activités de conception :
  - Mécanisme de détection et traitement d'erreur
  - Raffinement des invariants de sécurité
  - Choix de conception minimisant les risques
  - Identification des contraintes sur l'exploitation
  - Stratégie et moyens de diagnostic, reconfiguration, reprise
  - Identification des tests spécifiques

Actions en réduction de risque à mettre en œuvre (suite)

- Lors des activités de codage :
  - Développement formel
  - Règles de codage spécifiques
  - Programmation défensive
  - Inspection code et documentation

Actions en réduction de risque à mettre en œuvre (suite)

- Lors des activités de test :
  - Equipe de test indépendante
  - Objectifs de couverture de tests plus sévères
  - Campagne de tests dédiés SdF
  - Validation de la bonne mise en œuvre des mécanismes de SdF :
    - activation au bon moment,
    - résultat de l'activation conforme à l'attendu

Lien avec d'autres référentiels

- Modèles de maturité (processus) : CMMI, ISO 15504 (SPICE), ISO 20000 (ITIL)
  - Certains aspects peuvent être communs :
    - Gestion des exigences, Gestion de configuration, Ingénierie, Vérification Validation,
  - Les aspects management projet, estimation, amélioration des processus, … ne sont pas couverts par les normes qui traitent de la SdF
- Méthodes AGILE développement avec des spécifications incomplètes, mais si elle est mal mise en œuvre peut donner des logiciels peu fiables
- PMBok du PMI pour couvrir les aspects Projet, les développements logiciels devenant de plus en plus souvent de "gros" projets, de l'alternative AGILE

APPLICATION DE LA SDF AUX SYSTÈMES PROGRAMMÉS

– Techniques d’analyse

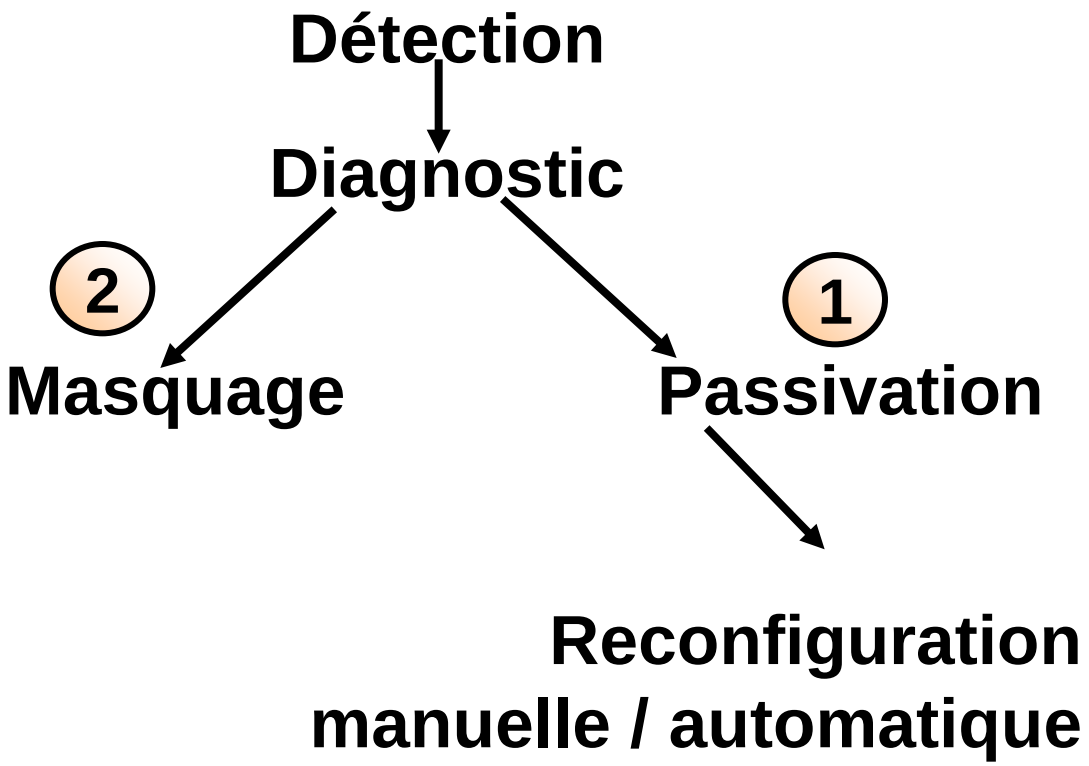
➔ Architectures pour la Sûreté de Fonctionnement

– Démarche de conception et de codage

– Moyens de vérification

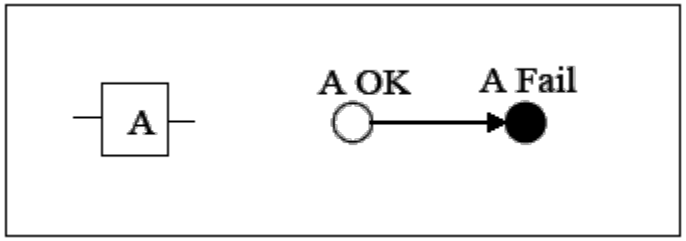
Approches possibles de la tolérance aux fautes

- 2 approches :
  - 1 : l'erreur est éliminée (préférable car conserve capacité de tolérance)
  - 2 : la conséquence de l'erreur est éliminée, l'erreur reste présente

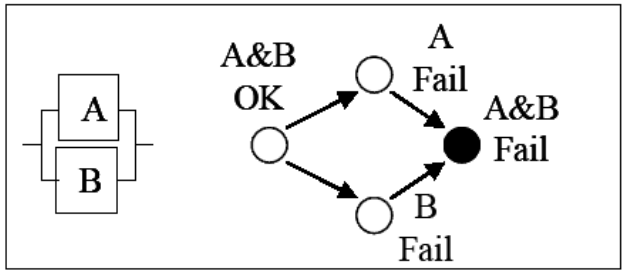


Comparaison de la capacité de tolérance aux fautes en fonction des architectures

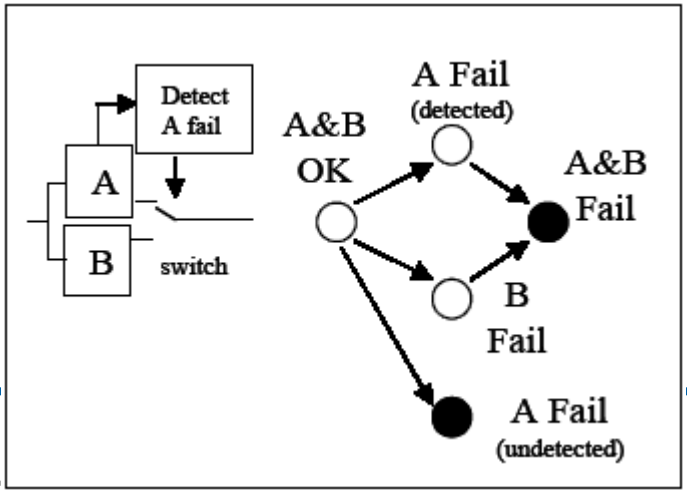
- Système simplex



- Système dual idéal : redondance active

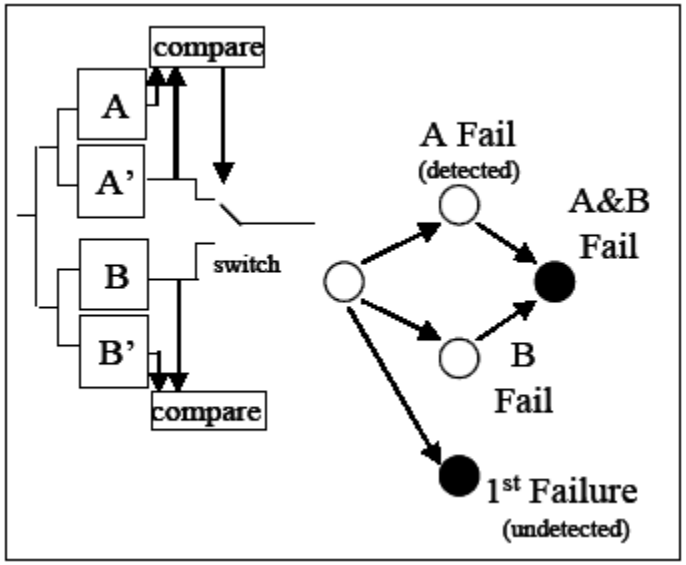


- Système dual (avec commutateur) : redondance passive

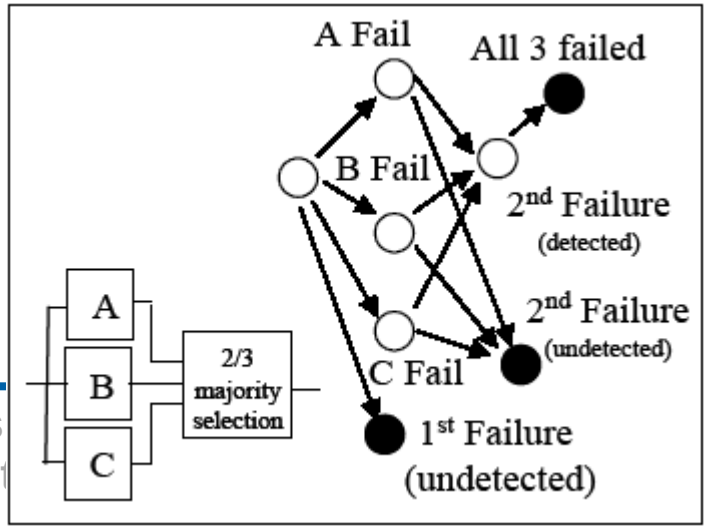


Comparaison de la capacité de tolérance aux fautes en fonction des architectures

- Système dual-dual actif/passif



- Système triplex



Architectures tolérantes aux fautes

- Détection en ligne d'erreurs logicielles :
  - tests d'acceptation des résultats et détection de la défaillance
- Bloc de recouvrement :
  - tests d'acceptation des résultats (directe ou indirecte) et essai d'une nouvelle solution si les résultats ne sont pas satisfaits
- Programmation en n versions :
  - exécution en parallèle de plusieurs solutions (distinctes ou identiques) et détection par comparaison relative ou absolue pour continuité de service
- Programmation en n autotestables :
  - architecture complexe basée sur composants redondés avec contrôle de résultats pour accroître les capacités de tolérance aux fautes du système

Comparaison des différentes architectures tolérantes aux fautes

Méthode	Technique de traitement		Jugement de l'acceptabilité des résultats	Schéma d'exécution des variantes	Cohérence des données d'entrée	Suspension du service durant le traitement	Variantes pour tolérer f fautes
Blocs de recouvrement	Détection d'erreur par test d'acceptation et reprise		Absolu, par rapport à la spécification	Séquentiel	Implicite par le principe de reprise	Oui, durée nécessaire à l'exécution d'une ou plusieurs variantes	*f* + 1
Programmation N-autotestable	Détection d'erreur et commutation des résultats	Détection par test d'acceptation		Parallèle	Explicite, par mécanismes dédiés	Oui, durée nécessaire à la commutation des résultats	
		Détection par comparaison	Relative, sur les résultats produits par les variantes				2(*f*+1)
Programmation N-versions	Vote					Non	*f* + 2

APPLICATION DE LA SDF AUX SYSTÈMES PROGRAMMÉS

- Techniques d'analyse
- Architectures pour la Sûreté de Fonctionnement
- ➔ Démarche de conception et de codage
- Moyens de vérification

Démarche de conception et de codage

- Méthode de conception :
  - Choix de méthodes orientées SdF : ex. Méthode B
- Mécanismes de tolérance aux fautes :
  - Eviter la propagation des défaillances (détection, signalisation),
  - Assurer la continuité de service (diversification)
  - Mise en place de mécanismes de SdF (tolérance, correction)
- Règles de Spécification / Conception / Codage
- Génération automatique de code

Mécanismes de tolérance aux fautes

- Détection des erreurs :
  - Trouver les équations, les indices, les inter-comparaisons, les contrôles de cohérence qui indiquent une erreur ou une probabilité d'erreur
- Traitement de l'erreur détectée :
  - Capacité à signaler l'erreur
  - Décidabilité de l'état et de l'action à entreprendre
  - Faisabilité technico-économique : cohérence avec les enjeux de sûreté
  - Compatibilité avec les ressources CPU, mémoire, …

Sécurisation de la conception

- Protection :
  - contre la modification intempestive de données internes :
    - Pour cause de bogue logiciel (écrasement de données, …)
    - Pour problème matériel mémoire
  - par des techniques de sécurisation :
    - Recopie d'une donnée critique en plusieurs endroits mémoire et comparaison des valeurs avant utilisation.
    - Utilisation de valeurs « complexes »
- Déterminisme du flot de contrôle :
  - Garantie de la logique de fonctionnement
  - Adaptation du comportement sur problème interne

APPLICATION DE LA SDF AUX SYSTÈMES PROGRAMMÉS

- Techniques d’analyse
- Architectures pour la Sûreté de Fonctionnement
- Démarche de conception et de codage

➔ Moyens de vérification

Vérification de la SdF

- Principes de vérification :
  - Objectifs :
    - Conforter, et si possible démontrer, les choix effectués
    - Vérifier l'efficacité des dispositions prises
    - Finaliser les principes liés à l'exploitation opérationnelle
  - Principes :
    - Activité permanente dans toutes les étapes / activités du cycle de vie.
    - Plus souvent basée sur recherche d'un niveau de confiance suffisant plus que sur une réelle démonstration.

Vérification SdF lors des phases de tests

- Objectifs de test pour la SdF :
  - Fonctionnalités / mécanismes de tolérance aux fautes
  - Mise en oeuvre correcte de ces mécanismes
  - Robustesse globale du logiciel (aux diverses sollicitations et dans la durée)
  - Efficacité / performance / endurance

Vérification de la vérification SdF lors des phases de tests

- Indépendance du processus de tests
  - Détermination des cas de tests et vérification des tests
- Pertinence des tests
  - Injection de fautes pour juger de l'efficacité des tests
- Suffisance des tests
  - Renforcement de la couverture de test

RÉFÉRENTIELS NORMATIFS POUR SÉCURITÉ DES SYSTÈMES PROGRAMMÉS

➔Norme générique CEI 61508 Ed.2

- Déclinaisons pour le ferroviaire, l'automobile, …
- Point de vue de l'aéronautique

CEI 61508 Ed.2 : Présentation

- Titre : Sécurité fonctionnelle des systèmes électriques / électroniques / électroniques programmables relatifs à la sécurité
- But : Fournir une approche générale de toutes les activités liées au cycle de vie de sécurité de systèmes destinés à exécuter des fonctions de sécurité lorsqu'ils comportent des dispositifs (E/E/PE)
- Objectif majeur : permettre l'élaboration de normes internationales spécifiques à chaque domaine d'application
- Objectif induit : permettre le développement de systèmes E/E/PE relatifs à la sécurité en l'absence éventuelles de normes internationales pour ce secteur d'application

CEI 61508 Ed.2 : Structure de la norme

- P1 : concepts, organisation, cycle de vie, documentation, justifications à apporter et définition des SIL*
- P2 : exigences portant sur le matériel et le système
- P3 : exigences portant sur le logiciel
- P4 : glossaire (définitions et abréviations)
- P5 : lignes directrices pour la détermination des SIL
- P6 : indications pour la mise en œuvre des parties 2 et 3
- P7 : méthodes (measures), description et bibliographie

* SIL : Safety Integrity Level

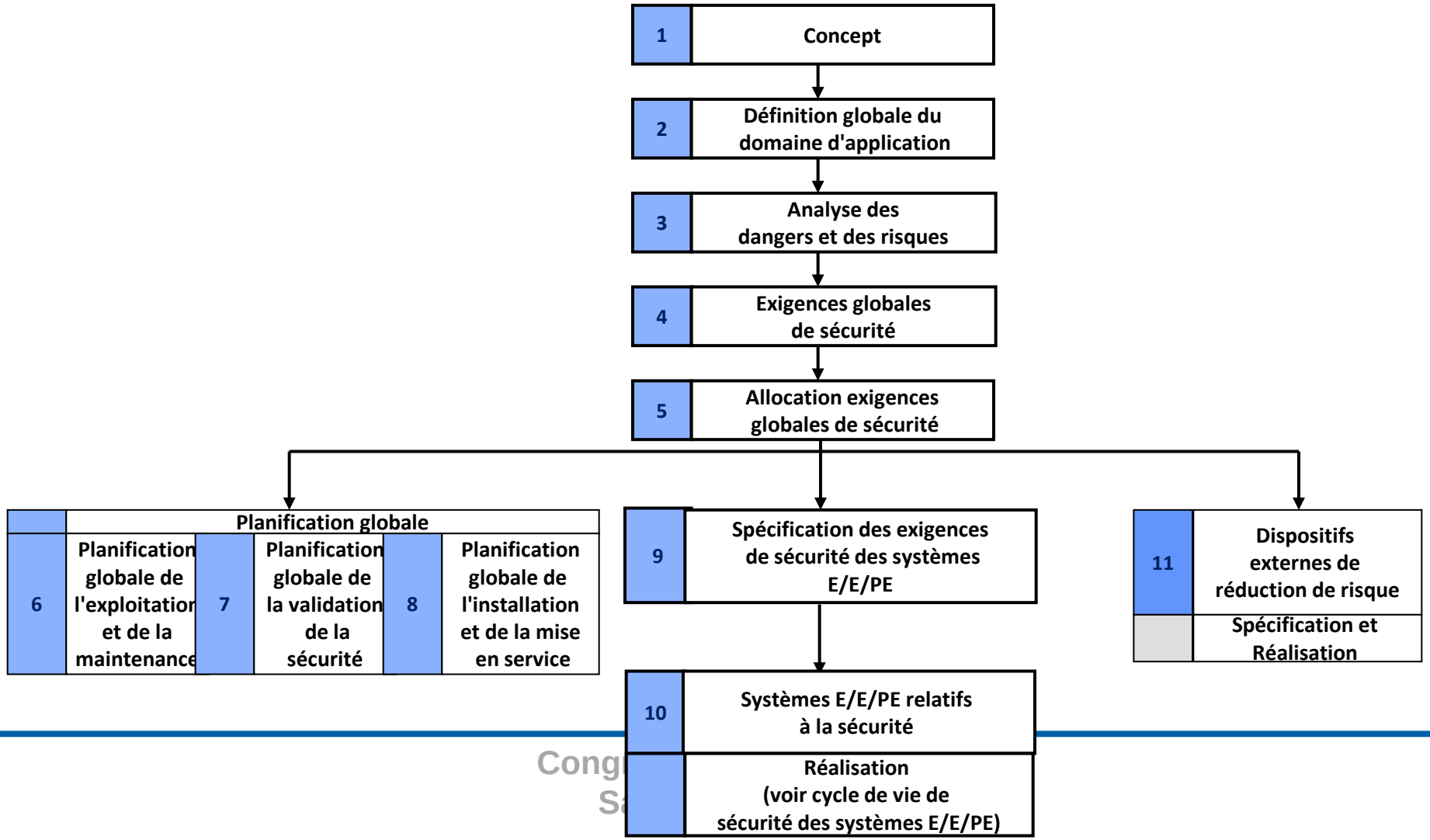
Être conforme à la CEI 61508 Ed. 2 c'est …

- Démontrer que toutes les prescriptions ont été remplies pour le niveau d'intégrité de sécurité spécifié :
  - mesures contre les défaillances systématiques
  - mesures contre les défaillances aléatoires :
    - contraintes architecturales
    - probabilité de défaillance
- Démontrer que ces prescriptions seront respectées durant toute la vie du système de sécurité

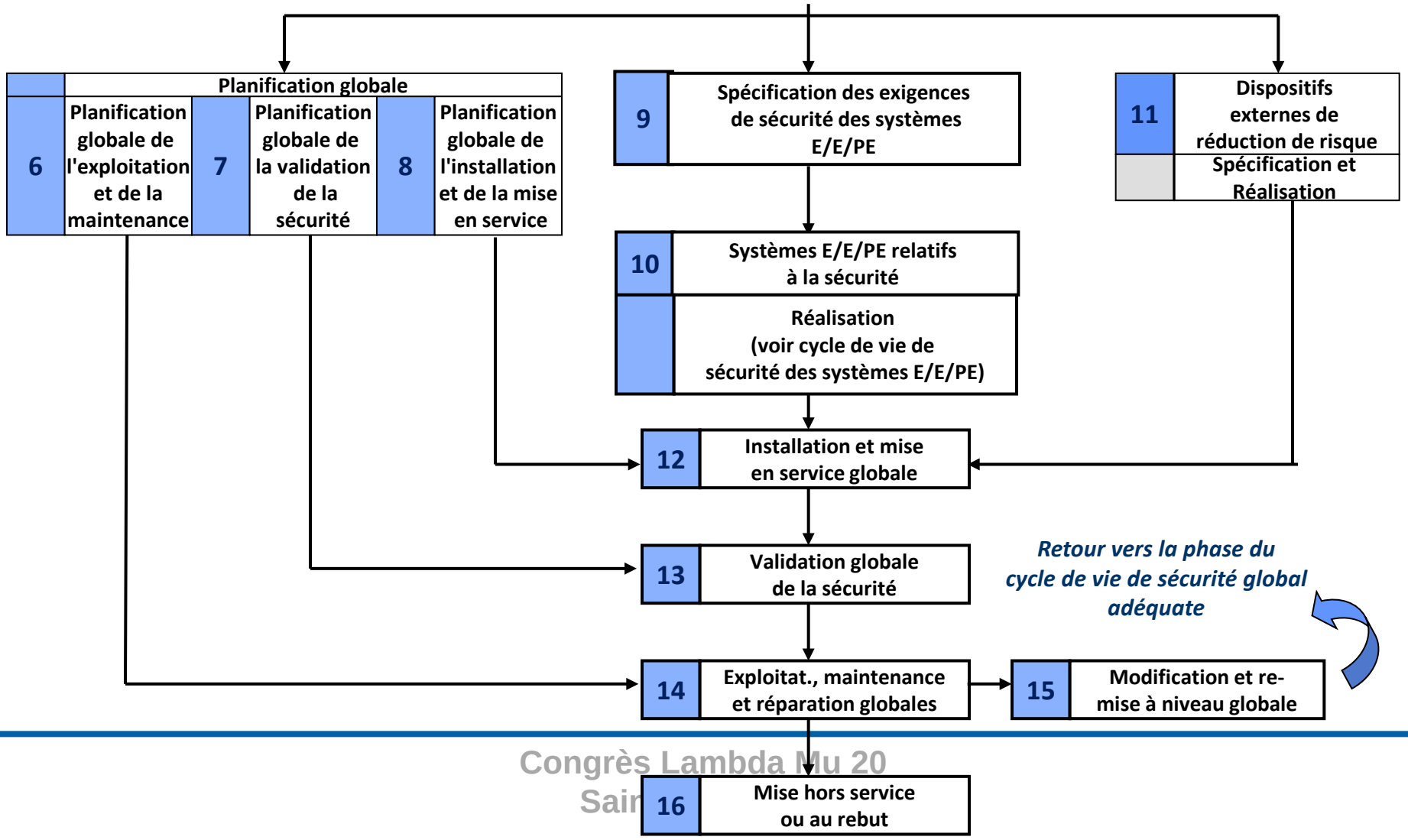
Une norme … des prescriptions

- Parmi les prescriptions de la norme :
  - certaines sont normalisées, d'autres sont informatives
  - certaines sont génériques quel que soit le niveau SIL de la fonctions de sécurité, d'autres dépendent du niveau de SIL requis
- Méthodes de démonstration ne sont pas imposées. La norme fournit des exemples de méthodes

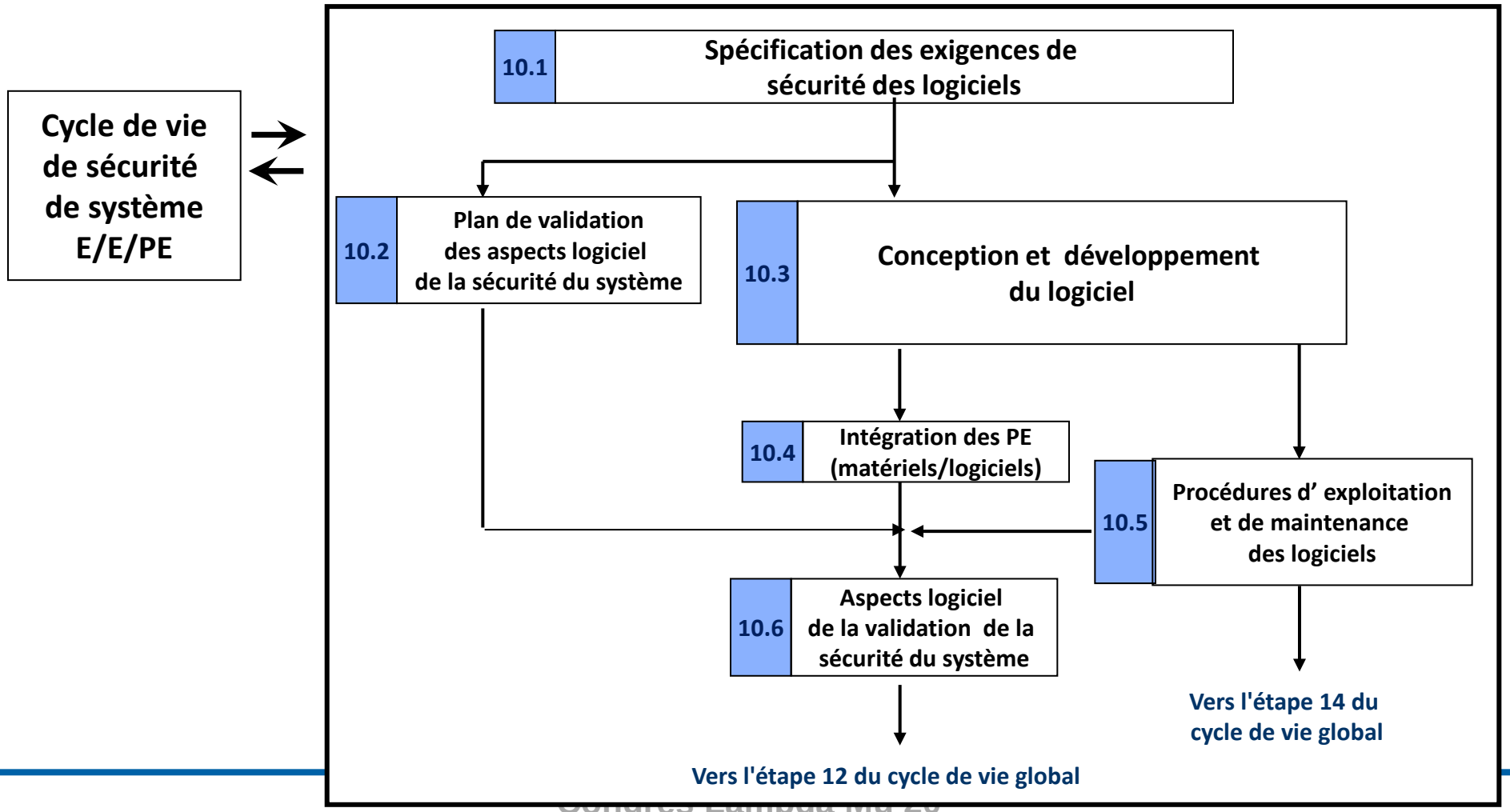
CEI 61508 Ed2 : Cycle de vie de sécurité global



CEI 61508 Ed2 : Cycle de vie de sécurité global



Cycle de réalisation des logiciels



CEI 61508 : Techniques et mesures en fonction du niveau de SIL (exemple)

- Tableau B.1 - Règles de conception et de codage

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Utilisation de règles de codage pour réduire la probabilité d'erreurs	C.2.6.2	HR	HR	HR	HR
2	Pas d'objets dynamiques	C.2.6.3	R	HR	HR	HR
3a	Pas de variables dynamiques	C.2.6.3	---	R	HR	HR
3b	Contrôle en ligne pendant la création de variables dynamiques	C.2.6.4	---	R	HR	HR
4	Utilisation limitée des interruptions	C.2.6.5	R	R	HR	HR
5	Utilisation limitée des pointeurs	C.2.6.6	---	R	HR	HR
6	Utilisation limitée de la récursion	C.2.6.7	---	R	HR	HR
7	Pas de branchements inconditionnels dans les programmes en langages de haut niveau	C.2.6.2	R	HR	HR	HR
8	Pas de conversion de type automatique	C.2.6.2	R	HR	HR	HR

RÉFÉRENTIELS NORMATIFS POUR SÉCURITÉ DES SYSTÈMES PROGRAMMÉS

– Norme générique NF EN 61508

➔Déclinaisons pour le ferroviaire, l'automobile, …

– Point de vue de l'aéronautique

Norme générique ➔ normes sectorielles

DO 178 C
DO 278 B
DO 30x
DO 254
Aéronautique

CEI 61508
NF EN61508
Norme générique

ECSS
NASA 871913

Spatial

CEI 61511
Process industriels

CEI 62061
Machines

CEI 61513
Nucléaire

EN 50126/8/9
Ferroviaire

ISO 26262
Automobile

ISO 13849
Machines

CEI 60780
CEI 60987
CEI 60880
CEI 62138

CEI 60601
CEI 62034
Médical

ISO25119
Tracteurs / matériels agricoles et forestiers

Lien direct ou indirect

Principes communs applicables au logiciel

- Adaptation des exigences à une cible de sécurité fonctionnelle (niveau d'intégrité de sûreté – SIL, ou équivalent)
- Détermination d'un ensemble de règles, techniques, méthodes, attendus selon le niveau visé
- Tous ne traitent que des aspects logiciels liés à la sécurité
- Evaluation indépendante de la sécurité obtenue
- Pas de résultat quantitatif attendu du niveau de SdF obtenu

RÉFÉRENTIELS NORMATIFS POUR SÉCURITÉ DES SYSTÈMES PROGRAMMÉS

- Norme générique NF EN 61508
- Déclinaisons pour le ferroviaire, l'automobile, …

➔Point de vue de l'aéronautique

DO 178 B/C (Aéronautique) : Principes fondamentaux

- Objectif d'application
  - Fournir aux acteurs de la communauté aéronautique un moyen reconnu de conformité aux règlements applicables
    - Objectifs
    - Moyens pour atteindre les objectifs
    - Exigences de démonstration de conformité DO 178 basée sur les informations produites :
      - Documentation de chaque activité
      - Vérification de chaque information
      - Traçabilité des informations
      - Couverture des informations par la vérification
- Démonstration de conformité aux règlements aéronautiques basée sur la qualité du développement du logiciel

DO 178 B/C : Contenu

- Couvre tous les processus du cycle de vie logiciel :
  - spécification,
  - conception,
  - codage,
  - intégration,
  - vérification,
  - assurance qualité,
  - gestion de configuration.
- Présente :
  - objectifs de ces processus,
  - activités pour satisfaire ces objectifs
  - preuves permettant de montrer que les objectifs ont été atteints.
- Le niveau logiciel engendre des exigences qui varient avec la catégorie de panne et donc avec la criticité des fonctions réalisées par le système.

DO 178 B/C : Vérification du logiciel

Exigences de haut niveau (Spécification)	Niveaux logiciel			
	A	B	C	D
Conformité / Exigences système	●	●	❍	❍
Exactitude et Cohérence	●	●	❍	❍
Précision des algorithmes	●	●	❍	-
Complétude - Traçabilité / Exigences Système	❍	❍	❍	❍
Respect des Standards liés aux exigences du Logiciel	❍	❍	❍	-
Testabilité	❍	❍	❍	-
Compatibilité par rapport à la machine cible	❍	❍	-	-

- ● : Vérification exigée avec indépendance
- ❍ : Vérification exigée
- \- : Pas de vérification exigée

CONCLUSION

Synthèse de la présentation

- Complexité du métier de la production de logiciel à haut niveau SdF
- Nécessité d’agir en parallèle sur les différents volets de la démarche :
  - Cycle de vie logiciel cohérent avec la logique système
  - Définir des architectures, identifier les mécanismes les plus utiles compte tenu du contexte
  - Renforcer les règles issues du REX à tous les niveaux de construction depuis l'expression de besoin, la spécification, la conception, le codage
- Fortes attentes sur la formalisation : choix, résultats et vérifications
- Nécessité niveau de rigueur élevé ("sanction" en cas de non respect)