Sûreté de fonctionnement

EISTI

Novembre 2016

Ali Baktash

Objectifs

- Comprendre la SDF et les enjeux
- Savoir quand appliquer une démarche Sûreté de fonctionnement
- Identifier les actions et les outils appropriés
- Connaitre des différentes normes SDF s'appliquant au logiciel

Agenda

1h30
- Introduction au SDF
- Rappel de la problématique pour le logiciel

1h30
- Tour d'horizon des différentes normes s'appliquant au logiciel

1h30
- Différentes approches pour améliorer la sûreté du logiciel
- Conclusion

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté du logiciel
- Conclusion

Pourquoi la SDF?

1. **Concevoir et fabriquer des systèmes sûrs** (fonctionnant ou se comportant tel que le client ou l'entreprise le souhaitent)
2. Détecter et traiter des écarts que l'on n'aurait pas vu en utilisant **les méthodes classiques de validation**
3. Détecter et traiter **plus tôt les écarts au moindre coût** (accroître la validation papier au détriment de la validation sur pièces réelles)
4. Utiliser les outils « qualité » de manière **cohérente et optimale**

Pourquoi la SDF?

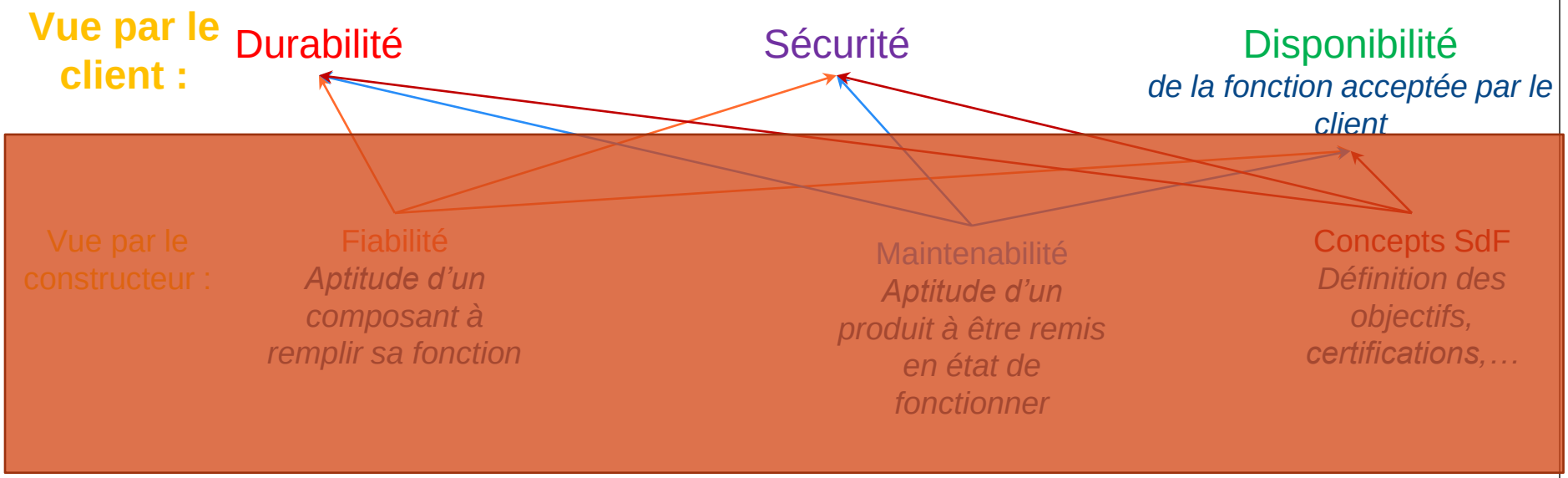
5. Être à même de **prouver l'utilisation de «l'état de l'art »** et des **« bonnes pratiques »**
6. **Dialoguer efficacement avec les fournisseurs** (description de notre besoin et confiance dans leurs réponses)
7. **Alimenter la base de connaissance** de l'Entreprise
8. Doter les produits d'un **avantage concurrentiel.**

→Diminution du coût d'obtention de la qualité

→Optimisation des risques et coûts juste et nécessaire

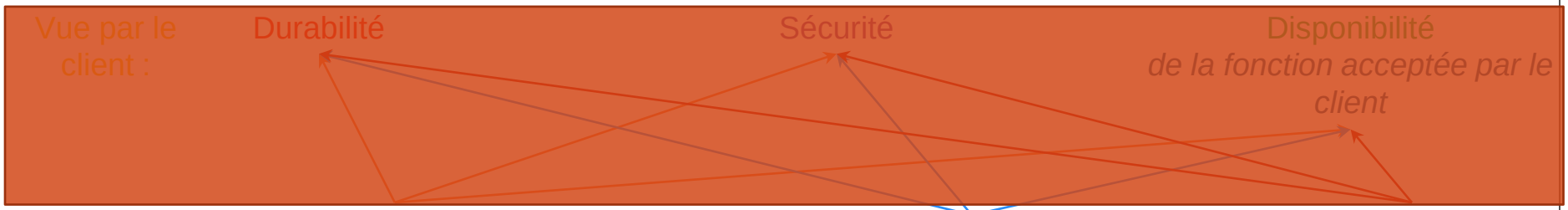
Définition SDF

Ensemble des aptitudes d'un produit lui permettant de disposer des performances fonctionnelles requises, au moment voulu, pendant la période voulue, sans dommage pour lui-même et pour son environnement.



Définition SDF

Ensemble des aptitudes d'un produit lui permettant de disposer des performances fonctionnelles requises, au moment voulu, pendant la période voulue, sans dommage pour lui-même et pour son environnement.



Vue par le constructeur :

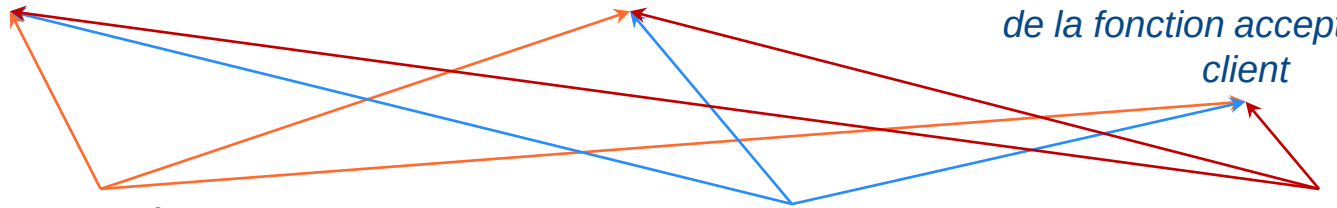
Fiabilité
Aptitude d'un composant à remplir sa fonction

Maintenabilité
Aptitude d'un produit à être remis en état de fonctionner

Concepts SdF
Définition des objectifs, certifications,…

Définition SDF

Ensemble des aptitudes d'un produit lui permettant de disposer des performances fonctionnelles requises, au moment voulu, pendant la période voulue, sans dommage pour lui-même et pour son environnement.

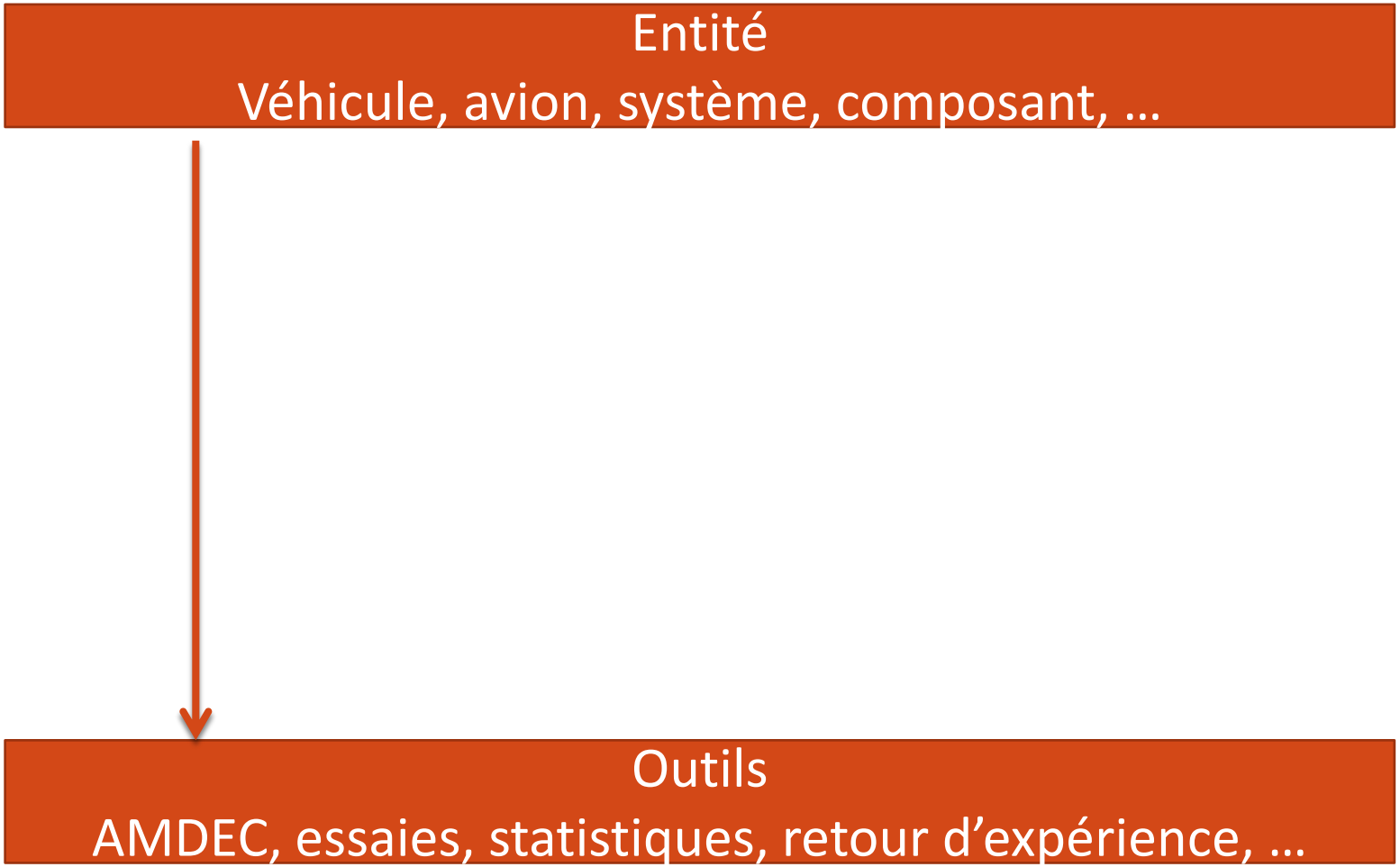


USA :	France :	UK :
Reliabilty	**F**iabilité	Dependability
Availability	**M**aintenabilité	
Maintenability	**D**isponibilité	
Safety	**S**écurité	

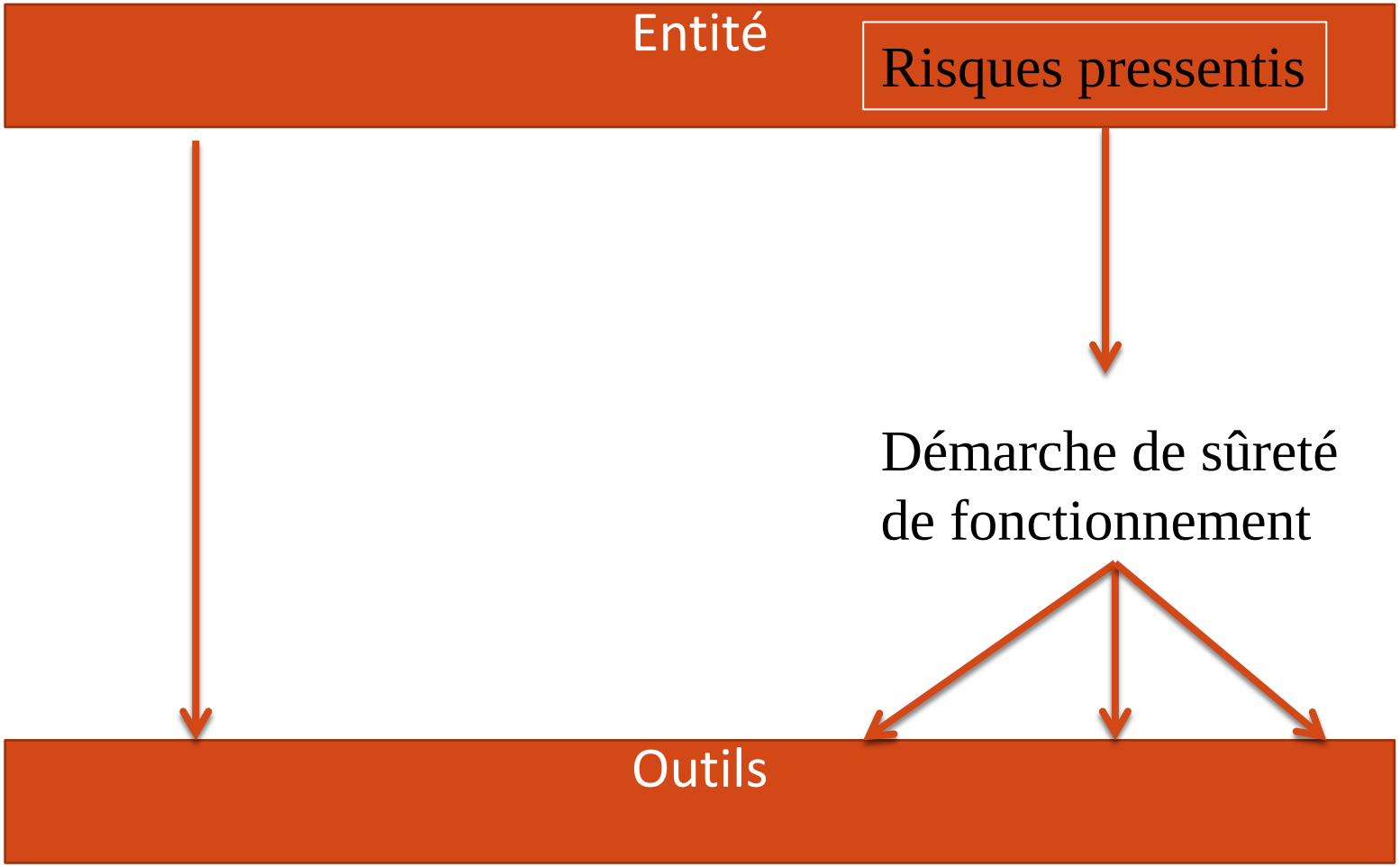
Autres définitions :

- **Système :** Ensemble de produits en interaction destiné à remplir une ou plusieurs fonctions pour le client.
- **Evènement indésirable :** Libellé des conséquences d'une défaillance aboutissant ou pouvant aboutir à un mécontentement client ou à une atteinte à l'environnement.
- **Risque :** Evènement dont l'apparition n'est pas certaine et dont la manifestation est susceptible d'engendrer des dommages significatifs.

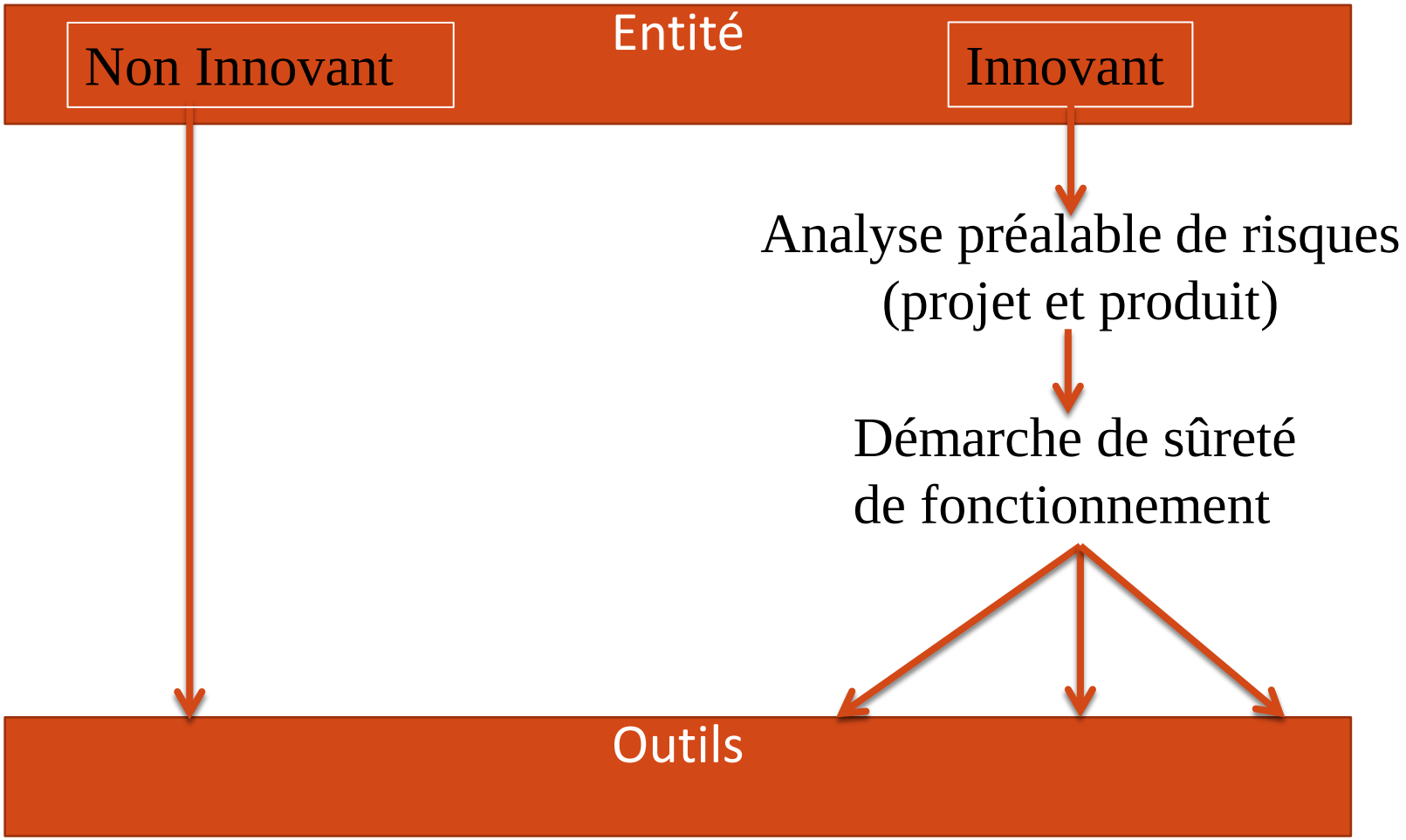
Le temps des outils



Le temps des balbutiements



Le temps de la SDF



Système classique et système innovant

Système classique:

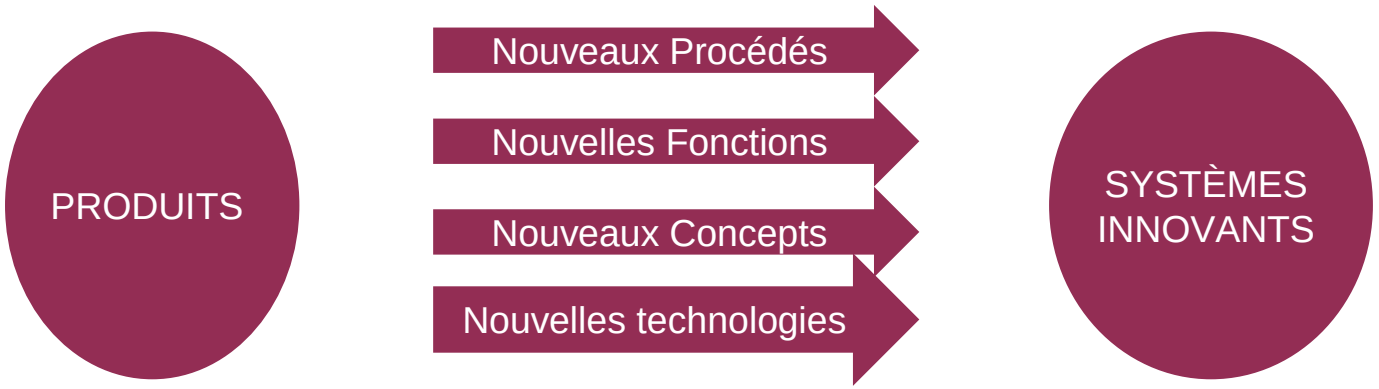
- Traitement par rapport à des « références »

  →Retour d'expérience à partir des données issues des mesures sur le terrain
- Complément par d'autres outils (AMDEC, Arbres de défaillances, etc…)
- → Est-ce suffisant aujourd'hui ?
  - Estimations prévisionnelles mais pas ou peu d'aide à la conception,
  - Fiabilité mais pas sécurité

Référence : système proche déjà en clientèle

Système classique et système innovant

Système innovant :



- → Traitement impossible par rapport à des « références »
- Aide à la conception mais pas d'estimations prévisionnelles

Référence : système proche déjà en clientèle

C’est quoi la SDF

- Démarche **systémique** qui met en évidence et traite les dysfonctionnements potentiels d’un système
  - Interdisciplinaire
  - Appréhender ce système, dans son environnement, dans son fonctionnement, dans ses mécanismes, dans ce qui n'apparait pas en faisant la somme de ses parties
- Démarche **TOP-Down** : part du client (utilisateur)
- 2 approches complémentaires:
  - Traitement des **RETEX**
  - Traitement du **risque potentiel**
- Démarche née dans l’aéronautique, l’espace et le nucléaire puis l’automobile qui couvre aujourd’hui les transports publics, les industries pétrolières et les industries grand public

Démarche SdF

APR- SdF

1. Définir ce qui risque d'arriver à l'utilisateur, au client (***effet client*** ou ***évènement redouté***)

- Mettre en évidence les évènements indésirables (**EI**)
- **Caractériser** les EI (gravité, scénarii d'apparition, etc…)

Plan de traitement

2. Allouer **des objectifs quantitatifs** et/ou **qualitatifs** par effet client

- Définir les **objectifs associés** (quantitatif et qualitatif)
- Choisir la **stratégie de traitement** (hiérarchiser, etc…)
- Définir une architecture qui assure **sécurité et disponibilité**

Traitement des EI

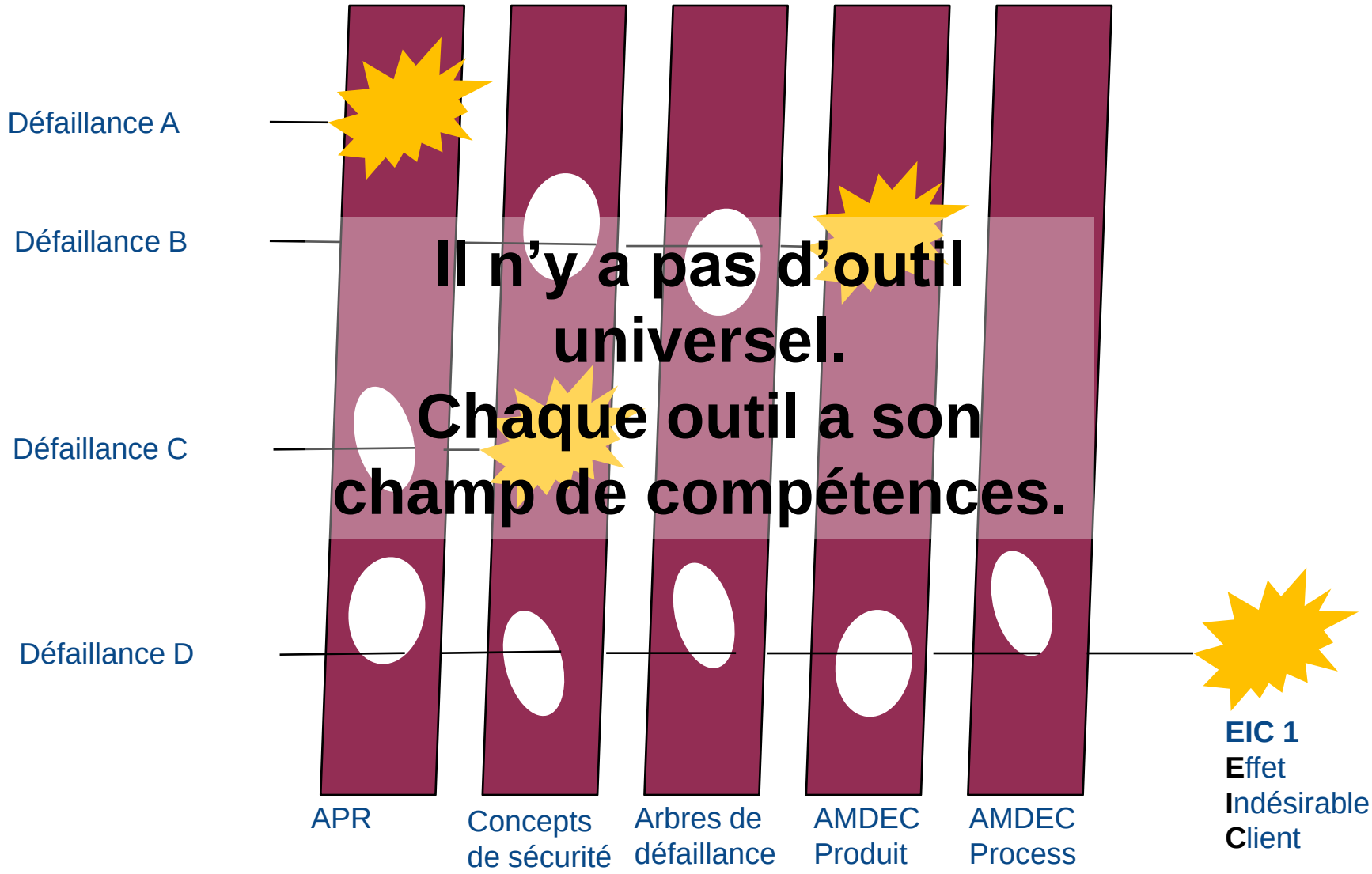
3. Chercher **les causes** pouvant aboutir à ces effets clients

- Mettre en œuvre des **analyses / méthodes de traitement** :
  - Arbres de défaillances par EI
  - Logique AMDEC
  - Analyse de zone, pire cas, etc…
  - Essais
  - Recommandations

4. Traiter ces causes de manière à **assurer les objectifs**

- **Gérer les risques**

Principe des tranches de gruyère



Phase d'application démarche SdF

Client :
- Projet

Clients :
- Bureau d'Etudes
- Projet

Clients :
- Projet
- Direction Qualité
- Bureau d'Etudes

Synthèse

- **Le plus tôt possible et tout au long du processus de conception :**
  - Mettre en évidence les évènements indésirables (dysfonctionnement du système),
  - Caractériser ces évènements indésirables,
  - Traiter ces évènements indésirables,
  - Apporter les éléments de décision.
  - Faire traiter les dysfonctionnements à la source,
  - Aider à la conception

- **Déléguer au fournisseurs dans le but d'utiliser la démarche pour :**
  - Choisir entre diverses solutions possibles,
  - Rendre plus sûr le fonctionnement des systèmes,
  - Bâtir la validation.

- **Nécessité de manager la Sûreté de Fonctionnement.**

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté du logiciel
- Conclusion

Problématique pour le logiciel

- Rappel définition SDF:

Aptitude à remplira les fonctions attendues en toutes circonstances sans mettre en péril la santé des utilisateurs et sans causer de dégâts pour lui-même et pour son environnement.

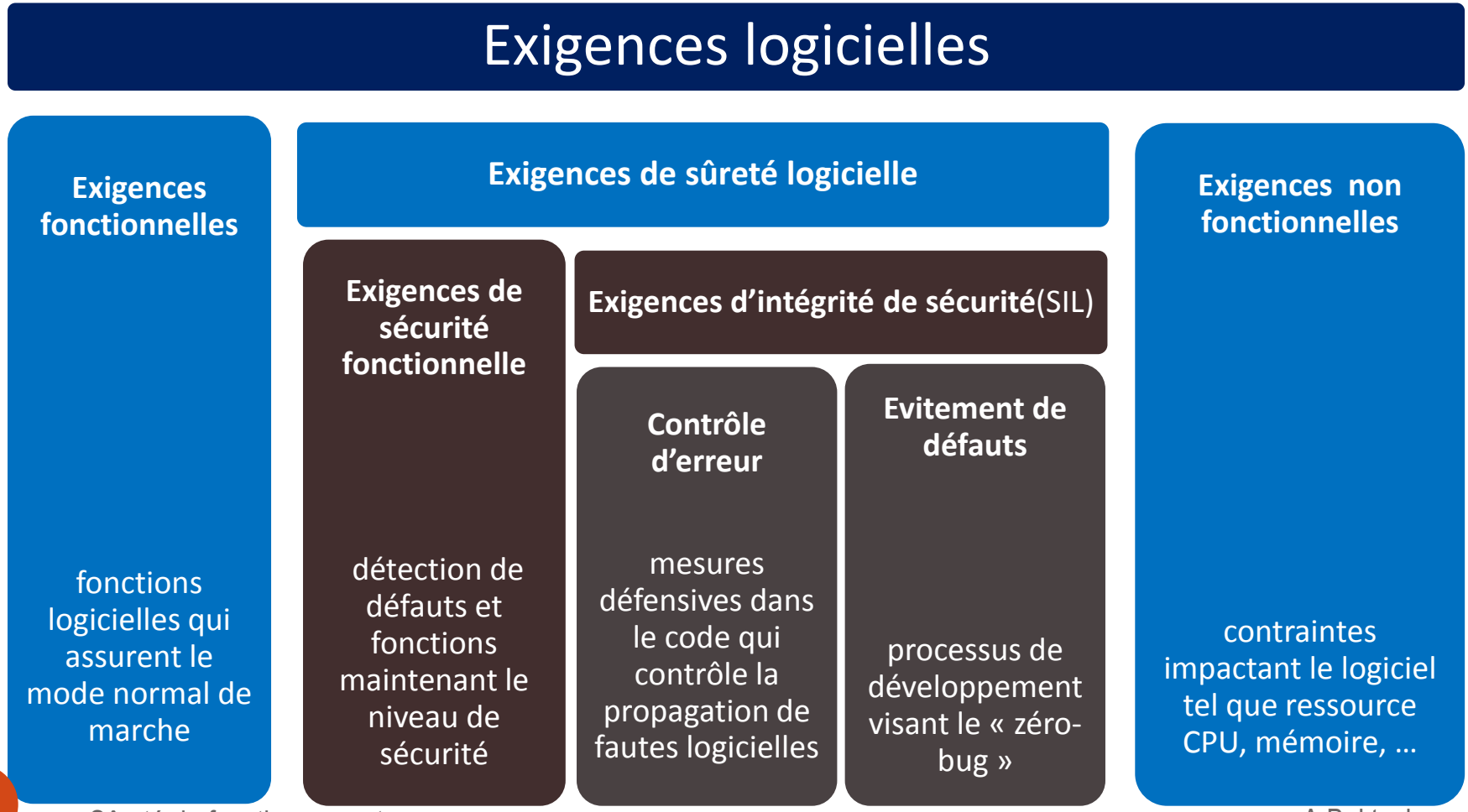
Problématique pour le logiciel

- Définition appliquée à un logiciel:

Utiliser des solutions techniques, des outils et des méthodes pendant tout le cycle de développement d'un logiciel permettant de s'assurer qu'il remplira les fonctions attendues en toutes circonstances sans mettre en péril la santé des utilisateurs , sans causer de dégâts pour lui-même et pour son environnement et sans causer des pertes financières.

Exigences sûreté de fonctionnement pesant sur le logiciel

Introduction et rappel de la problématique



Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté du logiciel
- Conclusion

Tour d'horizon des différentes normes SDF s'appliquant au logiciel

ISO 26262
Road Vehicles-FS

EN 50156
Electrical equipment for furnaces & ancillary equipment

ISO 15998
Earth Moving Machinery

IEC 62061
Safety of machinery

DO-178B
SW Considerations in Airborne Systems

IEC 61508
Functional safety of electrical/electronic Systems

IEC 60335
Safety of household & electrical appliances

EN 50128
Signaling and Processing Sys SW for Railway Control & Protection Systems

IEC 61511
Safety instrumented systems for the process industry sector

IEC 61513
Nuclear Power Plants

IEC 60601
Medical Electrical Equipment

Tour d'horizon des différentes normes SDF s'appliquant au logiciel

- **DO178C (aéronautique)**
- **CEI 61508-1 et CEI61508-3, et suite des normes CEI61508 (systèmes électroniques)**
- **ISO 26262 (automobile)**
- **IEC 60730 / 60335 (électroménager)**

Tour d'horizon des différentes normes SDF s'appliquant au logiciel

- **DO178C (aéronautique)**

- **CEI 61508-1 et CEI61508-3, et suite des normes CEI61508 (systèmes électroniques)**
- **ISO 26262 (automobile)**
- **IEC 60730 / 60335 (électroménager**

DO 178C

- Années 70: apparition du logiciel dans l'aéronautique

→ la certification des systèmes est devenu plus complexe, le système devant résister à des erreurs de conception, d'interaction logiciel et à d'éventuels disfonctionnements des composants matériels.

- L'utilisation unique des méthodes traditionnelles d'évaluation de sûreté basées sur des probabilités d'échec ne suffisait plus et il a été nécessaire d'introduire de nouvelles méthodes pour assurer le bon fonctionnement des systèmes embarquant du logiciel.

→ Création de DO-178 par RTCA en mai 1980 aux Etats Unis ainsi qu'à celle de ED-12 par EUROCAE en Europe.

RTCA: Radio Technical Commission for Avionics

EUROCAE: European Organization for Civil Aviation Equipment

Normes SDF s'appliquant au logiciel

1970's	Engineers discover software
1980	DO-178-A RTCA
1989	DO-178B/ED-12B RTCA/EUROCAE
1992	DO-178-B Final approval
2012	DO-178-C RTCA/EUROCAE G71

RTCA: Radio Technical Commission for Avionics

EUROCAE: European Organization for Civil Aviation Equipment

DO 178C

- 5 classes de logiciels
  - **Niveau A** : Un disfonctionnement provoquerait ou contribuerait à une situation catastrophique de perte d'appareil.
  - **Niveau B** : Un disfonctionnement provoquerait ou contribuerait à une situation dangereuse ou un disfonctionnement sévère et majeur d'appareil.
  - **Niveau C** : Un disfonctionnement provoquerait ou contribuerait à un disfonctionnement majeur de l'appareil. Ce niveau représente à peu près 85%[1] des niveaux A/B (en terme de nombres d'objectifs[2]).
  - **Niveau D** : Un disfonctionnement provoquerait ou contribuerait à un disfonctionnement mineur de l'appareil.
  - **Niveau E** : Aucun impact sur le fonctionnement de l'appareil ou la charge de travail du pilote.

[1] Ce pourcentage ne reflète en rien la charge nécessaire pour passer du niveau C au niveau A ou B. Cette charge dépend de nombreux critères spécifiques à chaque développement : il serait donc hasardeux de vouloir avancer un chiffre précis.

[2] La variation est surtout sensible sur le processus de conception, sur l'exigence de couverture structurelle des tests, et sur l'indépendance.

Normes SDF s'appliquant au logiciel

DO 178C

Table A-7

Verification Of Verification Process Results

	Objective		Applicability by SW Level				Output		Control Category by SW level			
	Description	Ref.	A	B	C	D	Description	Ref.	A	B	C	D
1	Test procedures are correct.	6.3.6b	●	○	○		Software Verification Cases and Procedures	11.13	②	②	②	
2	Test results are correct and discrepancies explained.	6.3.6c	●	○	○		Software Verification Results	11.14	②	②	②	
3	Test coverage of high-level requirements is achieved.	6.4.4.1	●	○	○	○	Software Verification Results	11.14	②	②	②	②
4	Test coverage of low-level requirements is achieved.	6.4.4.1	●	○	○		Software Verification Results	11.14	②	②	②	
5	Test coverage of software structure (modified condition/decision) is achieved.	6.4.4.2	●				Software Verification Results	11.14	②			
6	Test coverage of software structure (decision coverage) is achieved.	6.4.4.2a 6.4.4.2b	●	●			Software Verification Results	11.14	②	②		
7	Test coverage of software structure (statement coverage) is achieved.	6.4.4.2a 6.4.4.2b	●	●	○		Software Verification Results	11.14	②	②	②	
8	Test coverage of software structure (data coupling and control coupling) is achieved.	6.4.4.2c	●	●	○		Software Verification Results	11.14	②	②	②	

LEGEND:		
	●	The objective should be satisfied with independence.
	○	The objective should be satisfied.
	Blank	Satisfaction of objective is at applicant's discretion.
	①	Data satisfies the objectives of Control Category 1 (CC1).
	②	Data satisfies the objectives of Control Category 2 (CC2).

Tour d'horizon des différentes normes SDF s'appliquant au logiciel

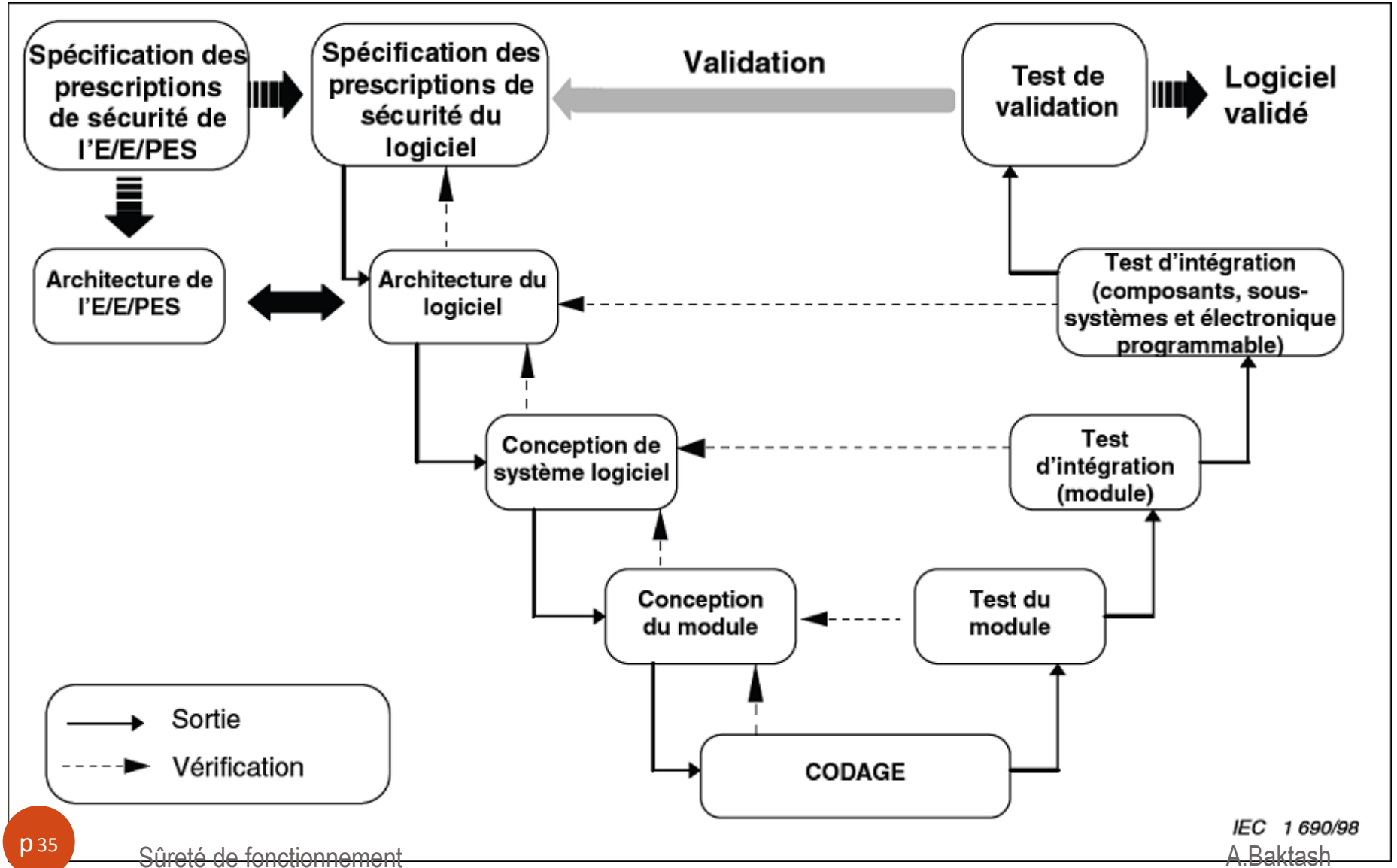
- **DO178C (aéronautique)**
- **CEI 61508-1 et CEI61508-3, et suite des normes CEI61508 (systèmes électroniques)**
- **26262 (automobile)**
- **IEC 60730 / 60335 (électroménager**

IEC 61508

- Cette norme est découpée en 7 parties :
  - Partie 1 : Exigences générales de conception (cycle de vie).
  - Partie 2 : Exigences concernant les systèmes électriques.
  - Partie 3 : Exigences concernant les logiciels.
  - Partie 4 : Définitions, terminologie.
  - Partie 5 : Guide pour la mise en œuvre de la partie 1 (Annexes informatives).
  - Partie 6 : Guide pour la mise en œuvre des parties 2 et 3 (Annexes informatives).
  - Partie 7 : Bibliographie des techniques (Annexes informatives).

Normes SDF s'appliquant au logiciel

IEC 61508



IEC 61508

- L'IEC 61508 est **orientée « performances ».**

- Par opposition aux normes dites déterministes et prescriptives, c'est l'utilisateur qui, à travers son analyse et son évaluation du risque détermine les performances à atteindre par son système E/E/PE concerné par la sécurité.

- La norme spécifie les performances correspondant au risque déterminé et les moyens (techniques et mesures) à mettre en œuvre pour garantir l'obtention des performances du système.

E/E/PE: Electrical/Electronic/Programmable Electronic Safety-related Systems

IEC 61508

- Approche basée sur le risque pour déterminer les exigences **d'intégrité** de la **sécurité** des systèmes E/E/PE concernés,
  - Fournit des exemples de manières à y parvenir.
- Modèle global de cycle de vie : Cadre technique pour les activités nécessaires afin de garantir que la sécurité fonctionnelle est atteinte par les systèmes E/E/PE concernés.
- Activités du cycle de vie de la sûreté
  - Conception initiale
  - L'analyse et l'évaluation des risques
  - Développement des exigences de sûreté, spécification, conception, implémentation
  - Exploitation, maintenance, modification
  - Jusqu'au démantèlement final et à la mise au rebut.

E/E/PE: Electrical/Electronic/Programmable Electronic Safety-related Systems

IEC 61508

- Pris en compte des aspects système (y compris tous les sous-systèmes réalisant des fonctions de sûreté, matériel et logiciel) ainsi que les mécanismes de pannes (matérielles aléatoires et systématiques).
- Exigences pour se prémunir des pannes (évitant l'introduction de défauts) et pour contrôler les pannes (garantissant la sécurité en présence de pannes).
- Spécification de techniques et mesures nécessaires pour atteindre le niveau d'intégrité de la sécurité nécessaire.

IEC 61508

- 4 niveaux d'intégrité (SIL – Safety Integrity Level)

Niveau d'intégrité de sécurité	Niveaux de Sécurité Probabilité annuelle de défaillance de la fonction	
	Fonctionnement en continu	Fonctionnement à la demande
	Taux de défaillance horaire	Probabilité de défaillance à la sollicitation
SIL 4	$10^{-8} < \lambda < 10^{-9}$	$10^{-4} < PFD_{avg} < 10^{-5}$
SIL 3	$10^{-7} < \lambda < 10^{-8}$	$10^{-3} < PFD_{avg} < 10^{-4}$
SIL 2	$10^{-6} < \lambda < 10^{-7}$	$10^{-2} < PFD_{avg} < 10^{-3}$
SIL 1	$10^{-5} < \lambda < 10^{-6}$	$10^{-1} < PFD_{avg} < 10^{-2}$

Normes SDF s'appliquant au logiciel
IEC 61508

Tableau A.5 – Conception et développement du logiciel: test et intégration des modules logiciels (voir 7.4.7 et 7.4.8)

	Technique/Mesure*	Réf.	SIL1	SIL2	SIL3	SIL4
1	Test probabiliste	C.5.1	---	R	R	HR
2	Analyse dynamique et test	B.6.5 Tableau B.2	R	HR	HR	HR
3	Enregistrement et analyse de données	C.5.2	HR	HR	HR	HR
4	Tests fonctionnel et boîte noire	B.5.1 B.5.2 Tableau B.3	HR	HR	HR	HR
5	Modélisation du fonctionnement	C.5.20 Tableau B.6	R	R	HR	HR
6	Test d'interface	C.5.3	R	R	HR	HR

NOTE 1 – Le test et l'intégration des modules logiciels sont des activités de vérification (voir tableau A.9).

NOTE 2 – Des techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

* Une technique/mesure repérée par un nombre doit être sélectionnée en fonction du niveau d'intégrité de sécurité.

R = Recommandé
HR = Hautement Recommandé
NR = Non Recommandé

Normes SDF s'appliquant au logiciel

IEC 61508

Tableau B.2 – Analyse dynamique et test
(référencés dans les tableaux A.5 et A.9)

	Technique/Mesure*	Réf.	SIL1	SIL2	SIL3	SIL4
1	Exécution de cas de test à partir de l'analyse des valeurs aux limites	C.5.4	R	HR	HR	HR
2	Exécution de cas de test à partir de l'estimation des erreurs	C.5.5	R	R	R	R
3	Exécution de cas de test à partir de l'implantation d'erreurs	C.5.6	---	R	R	R
4	Modélisation du fonctionnement	C.5.20	R	R	R	HR
5	Classes d'équivalence et test des partitions d'entrée	C.5.7	R	R	R	HR
6	Tests basés sur la structure	C.5.8	R	R	HR	HR

NOTE – L'analyse des cas de test se fait au niveau du sous-système et est basée sur la spécification et/ou la spécification et le code.

* Des techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

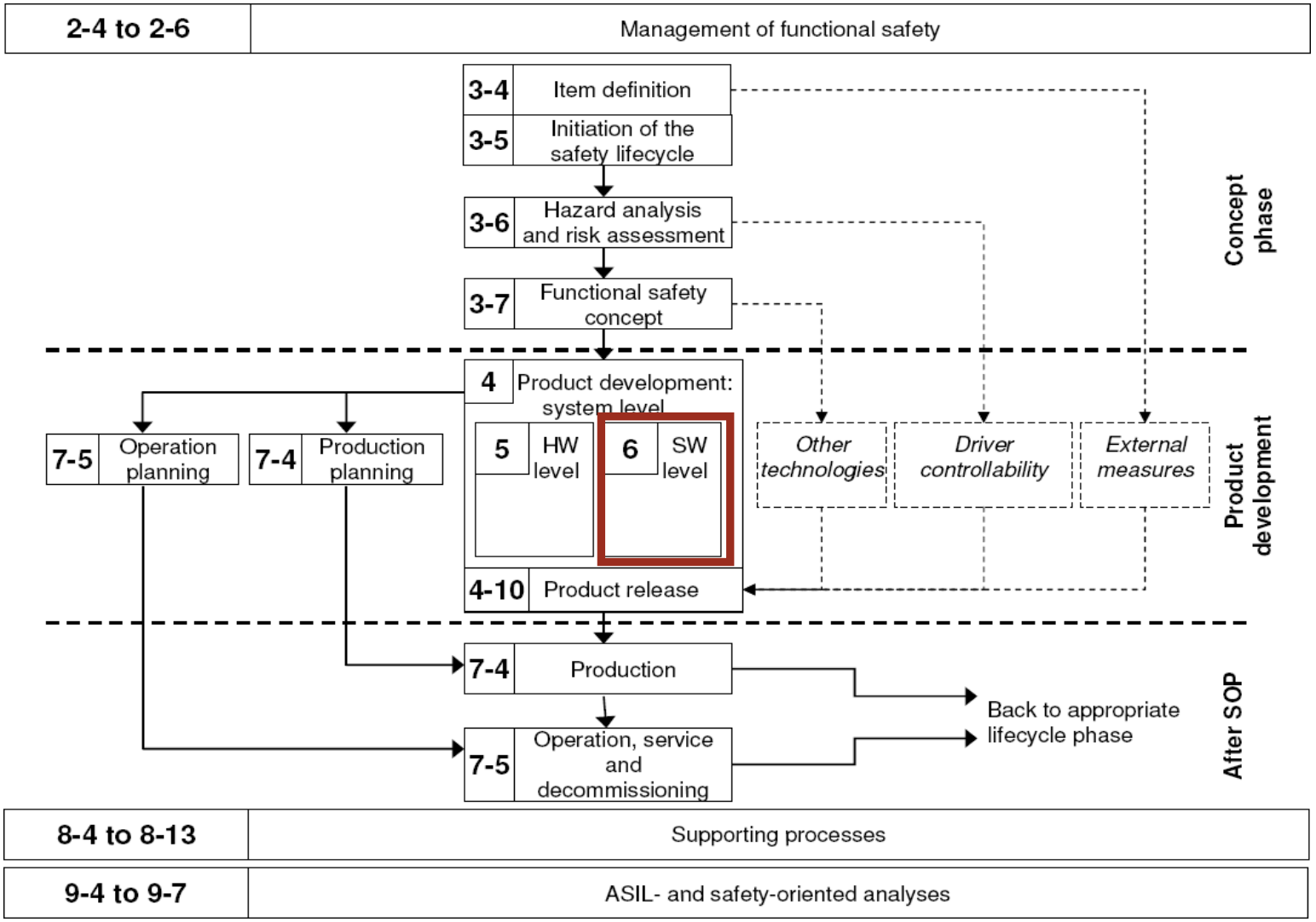
R = Recommandé
HR = Hautement Recommandé
NR = Non Recommandé

Tour d'horizon des différentes normes SDF s'appliquant au logiciel

- **DO178C (aéronautique)**
- **CEI 61508-1 et CEI61508-3, et suite des normes CEI61508 (systèmes électroniques)**
- **ISO 26262 (automobile)**
- **IEC 60730 / 60335 (électroménager)**

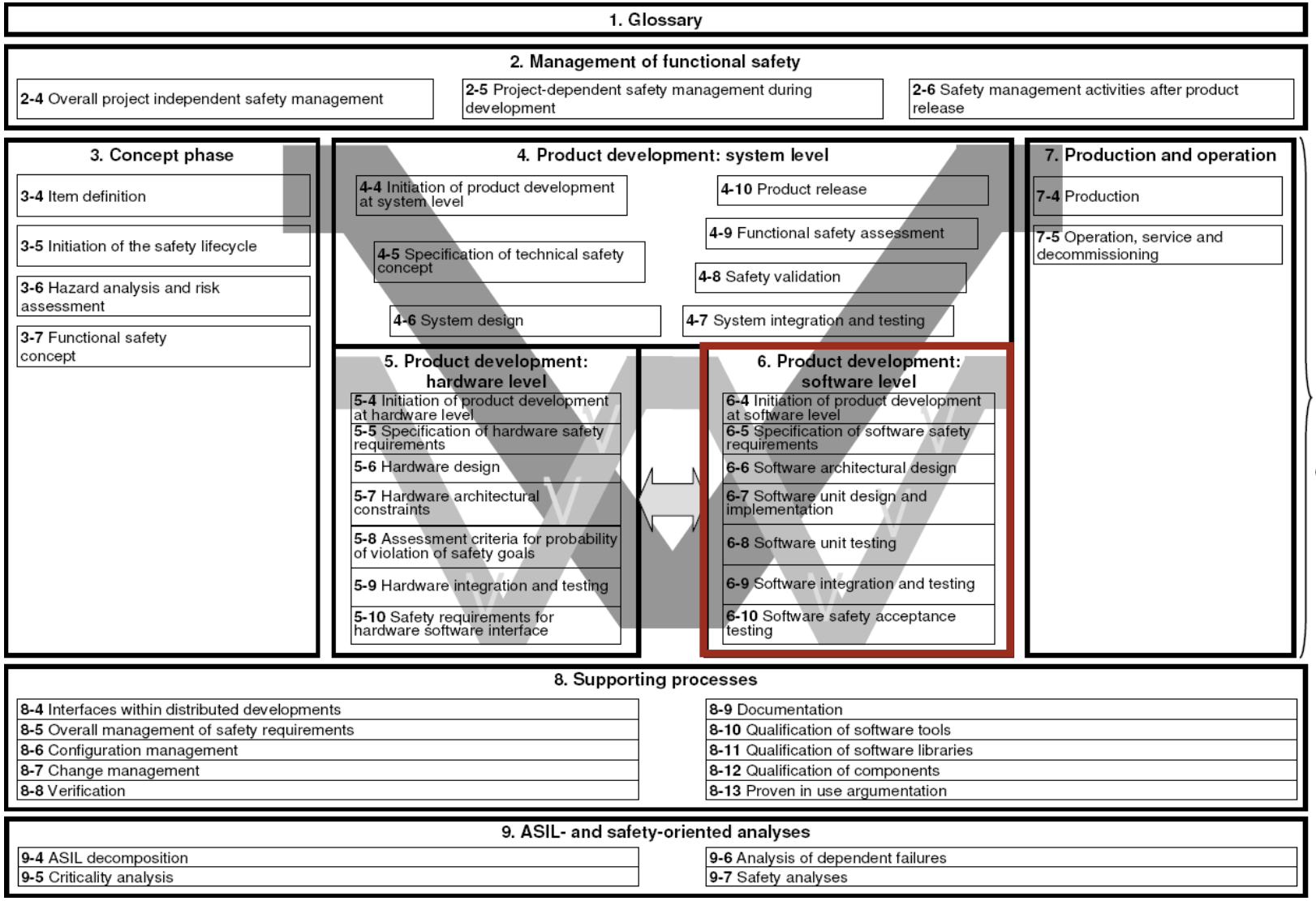
Normes SDF s'appliquant au logiciel

ISO 26262



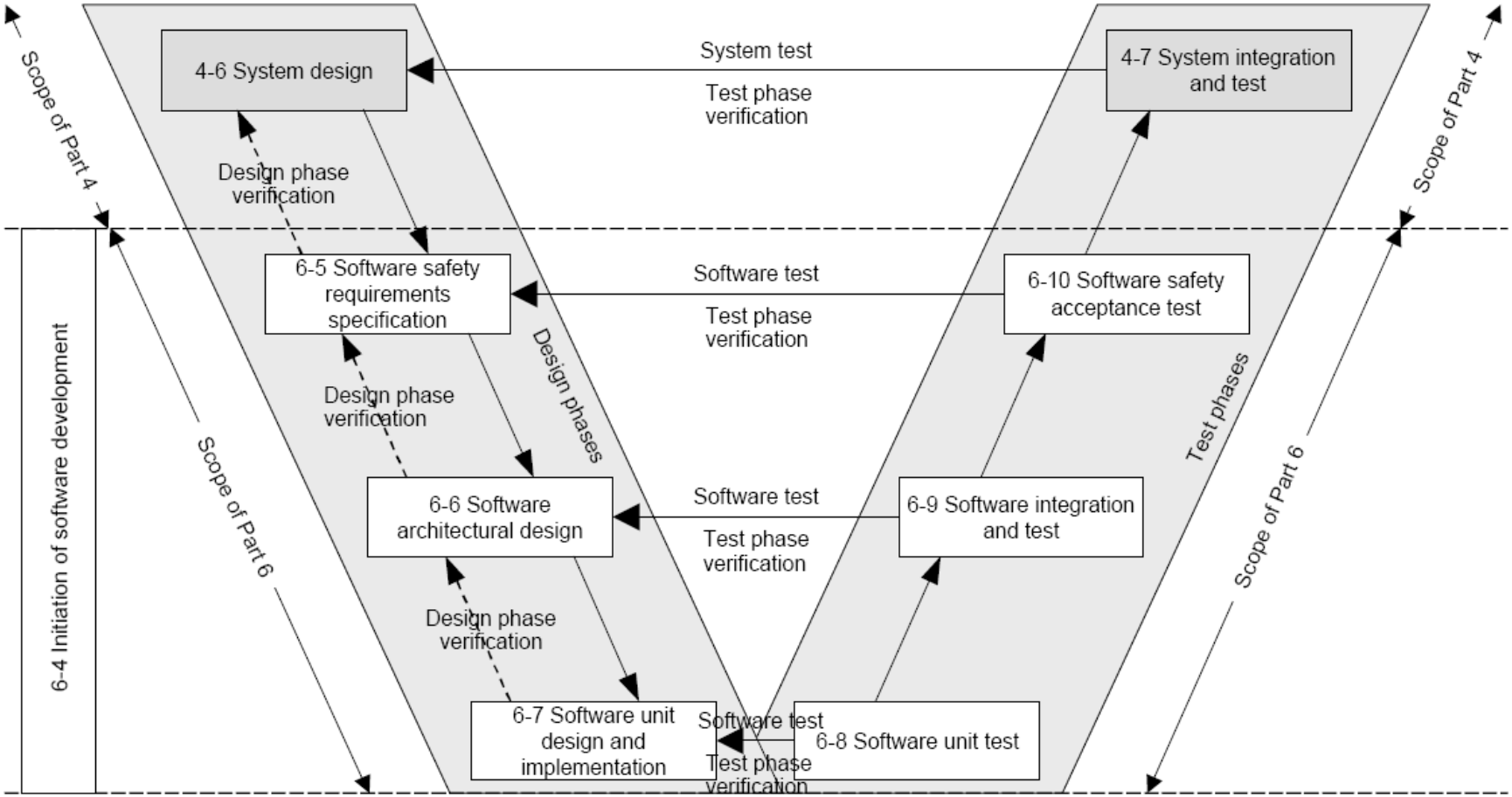
Normes SDF s'appliquant au logiciel

ISO 26262



Normes SDF s'appliquant au logiciel

ISO 26262



Reference phase model for the software development

ISO 26262

- 4 niveaux d'ASIL (A → D)

Controllability	Exposure	Severity			
		S0	S1	S2	S3
C1	E1	QM	QM	QM	QM
	E2	QM	QM	QM	QM
	E3	QM	QM	QM	A
	E4	QM	QM	A	B
C2	E1	QM	QM	QM	QM
	E2	QM	QM	QM	A
	E3	QM	QM	A	B
	E4	QM	A	B	C
C3	E1	QM	QM	QM	A
	E2	QM	QM	A	B
	E3	QM	A	B	C
	E4	QM	B	C	D

Normes SDF s'appliquant au logiciel

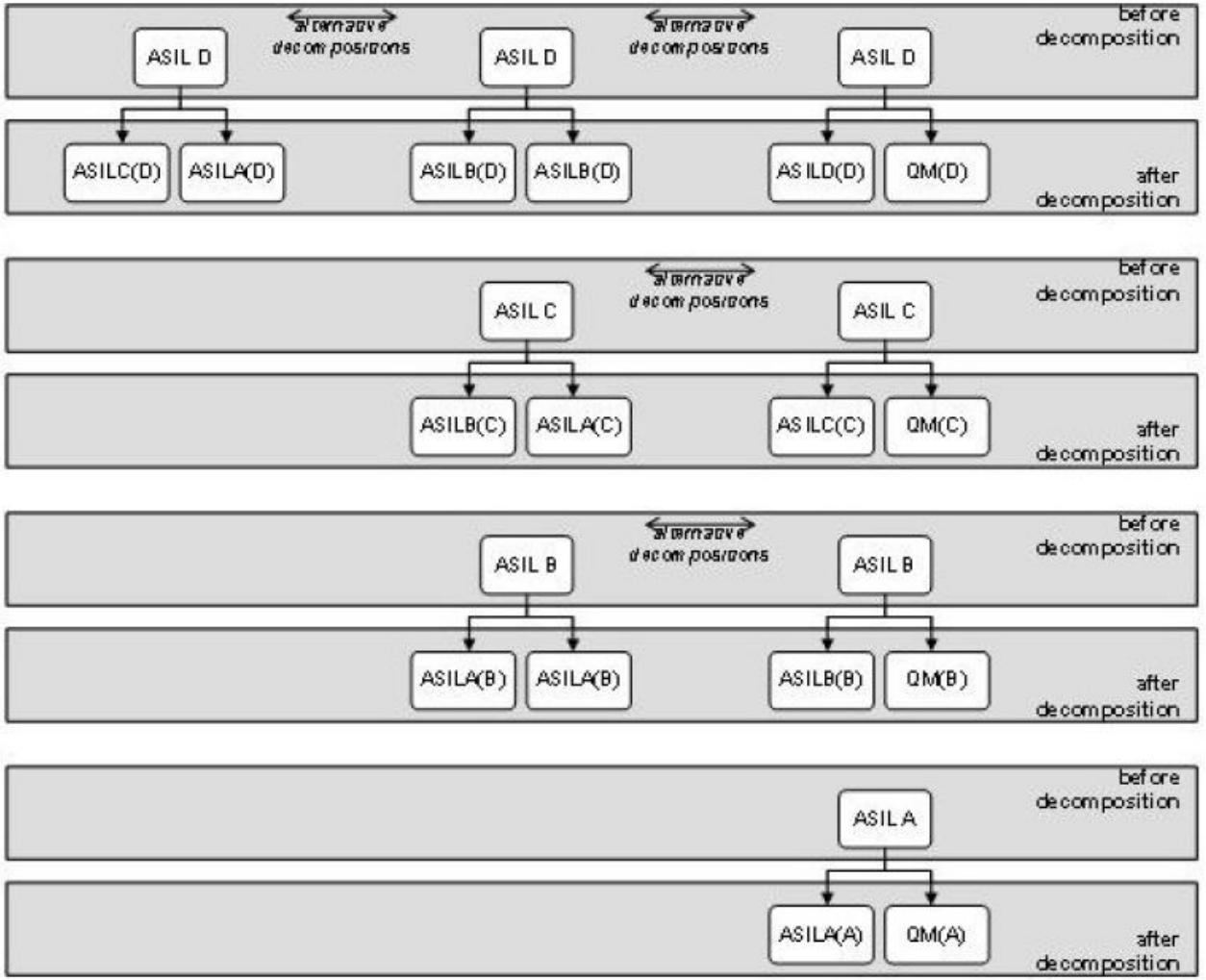
ISO 26262

S0	S1	S2	S3
No injuries	Light and moderate injuries	Severe injuries, possibly life-threatening, survival probable	Life-threatening injuries (survival uncertain) or fatal injuries
AIS 0 and less than 10% probability of AIS 1-6	more than 10% probability of AIS 1-6 (and not S2 or S3)	more than 10% probability of AIS 3-6 (and not S3)	more than 10% probability of AIS 5-6

E0	E1	E2	E3	E4
Incredible	Very low probability	Low probability	Medium probability	High probability
Never	Less often than once a year	A few times a year	Once a month or more often	Almost every drive
0%	Not specified	<1% of average operating time	1%-10% of average operating time	>10%

C0	C1	C2	C3
Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable
	99% or more of all drivers	90% or more of all drivers	Less than 90% of all drivers

La décomposition du niveau d'ASIL



Normes SDF s'appliquant au logiciel

ISO 26262

Methods and Measures		According to req.	ASIL			
			A	B	C	D
1a	Inspection of software safety requirements	5.4.8	++	++	++	++
1b	Walkthrough of software safety requirements	5.4.8	++	+	o	o
1c	Model review	5.4.8	++	++	++	++
3a	Formal verification	5.4.8	+	+	++	++
3b	Semi-formal verification	5.4.8	+	+	++	++

NOTE 1: The full set of software safety requirements needs to be verified by an appropriate combination of methods 1x. This combination typically depends on the applied methods and measures for specifying software safety requirements (see table 5.1)

NOTE 2: Method 3a requires a formal specification of the software safety requirements.

NOTE 3: In method 3b, simulation of software safety requirements is regarded as a semi-formal verification. This requires a specification of the software safety requirements by executable models.

Methods and Measures		According to req.	ASIL			
			A	B	C	D
1	Documentation of software architectural design in natural language	6.4.1	++	++	++	++
2a	Formal methods for software architectural design	6.4.1	+	+	++	++
2b	Semiformal methods for software architectural design	6.4.1	+	++	++	++
2c	Informal methods for software architectural design	6.4.1	++	++	+	+
3	Computer-aided tools for software architectural design	6.4.1	+	+	++	++
4	Guidelines for software architectural design	6.4.1	+	++	++	++

"++" The method or measure is highly recommended for this ASIL. If this technique or measure is not used then the rationale behind not using it shall be detailed.

"+" The method or measure is recommended for this ASIL.

"o" The method or measure has no recommendation for or against being used.

Tour d'horizon des différentes normes SDF s'appliquant au logiciel

- **DO178C (aéronautique)**
- **CEI 61508-1 et CEI61508-3, et suite des normes CEI61508 (systèmes électroniques)**
- **ISO 26262 (automobile)**
- **IEC 60730 / 60335 (électroménager)**

Normes SDF s'appliquant au logiciel

IEC 60730 / 60335

- 3 classes de logiciels

H.2.21 Définitions relatives à la classification des logiciels

H.2.21.1
logiciel de classe A
fonctions de commande qui ne sont pas destinées à avoir une influence sur la sécurité du matériel

Les exemples sont les thermostats d'ambiance, les capteurs d'humidité, les dispositifs d'allumage, les temporisateurs et interrupteurs temporisés.

H.2.21.2
logiciel de classe B
fonctions de commande destinées à prévenir les opérations non sécurisées du matériel commandé

Les exemples sont les coupe-circuit thermiques et les serrures de portes pour matériel de blanchisserie.

H.2.21.3
logiciel de classe C
fonctions de commande destinées à prévenir les risques spéciaux (par exemple explosion du matériel commandé)

Les exemples sont les dispositifs automatiques des brûleurs et les coupe-circuit thermiques pour les systèmes de chauffage de l'eau en circuit fermé (non ouverts).

IEC 60730 / 60335

H.11.12.2 Les dispositifs de commande avec fonctions déclarées comme logiciels de classe C doivent avoir une des structures suivantes:

– simple voie avec auto-test périodique et contrôle (H.2.16.7);

– deux voies (homogènes) avec comparaison (H.2.16.3);

– deux voies (différentes) sans comparaison (H.2.16.2).

La comparaison entre les structures deux voies peut être faite par:

– utilisation d'un comparateur (H.2.18.3) ou

– comparaison réciproque (H.2.18.15).

Les dispositifs de commande avec fonctions déclarées comme logiciels de classe B doivent avoir une des structures suivantes:

– simple voie avec test fonctionnel (H.2.16.5);

– simple voie avec auto-test périodique (H.2.16.6);

– double voie avec comparaison (H.2.16.1).

Les structures de logiciels de classe C sont aussi acceptables pour les dispositifs de commande à logiciels de classe B.

Normes SDF s'appliquant au logiciel

IEC 60730 / 60335

H.2.16.5
simple voie avec test fonctionnel
structure simple voie dans laquelle un test de données est introduit dans l'unité fonctionnelle avant son utilisation

H.2.16.6
simple voie avec auto-test périodique
structure simple voie dans laquelle les composants du dispositif de commande sont périodiquement testés au cours du fonctionnement

H.2.16.7
simple voie avec auto-test périodique et contrôle
structure simple voie avec auto-test périodique dans laquelle des moyens indépendants, capables chacun de fournir une réponse déclarée, contrôlent des aspects tels que les temporisations de sécurité, les séquences et le fonctionnement de logiciels

Normes SDF s'appliquant au logiciel

IEC 60730 / 60335

H.2.16.1
deux voies
structure comportant deux moyens fonctionnels mutuellement indépendants pour exécuter des manoeuvres spécifiées

Pour les dispositifs de commande ayant un mode commun défaut/erreur, une disposition spéciale peut être prise. Il n'est pas exigé que les deux voies soient chacune de nature algorithmique ou logique.

H.2.16.2
deux voies (différentes) avec comparaison
structure deux voies comportant deux moyens fonctionnels différents et mutuellement indépendants, chacun capable de fournir une réponse déclarée, et où est effectuée une comparaison des signaux de sortie pour une reconnaissance défaut/erreur

H.2.16.3
deux voies (homogènes) avec comparaison
structure deux voies comportant deux moyens fonctionnels identiques et mutuellement indépendants, chacun capable de fournir une réponse déclarée, et où est effectuée une comparaison des signaux internes ou des signaux de sortie pour une reconnaissance défaut/erreur

Normes SDF s'appliquant au logiciel

Quality Standards
- ISO 9001 ISO/TS 16949
- Caractéristique SW: ISO/CEI 9126
- C-ANSI ISO 9899
- ISO 25000/SQuaRE : ISO 9126 et 14598

Engineering Standards

ISO 12207 (SW lifecycle processes)
ISO 15288 (System lifecycle processes)

Assessment Models
- ISO 15504 (SPICE)
- CMMI

Safety Standards
- IEC 61508 (Meta-Standard)
- ISO TR 15497: MISRA Guidelines
- ECSS-E-40A (EU, Space)
- RTCA DO-178B (Aerospace SW, V&V)
- SAE APR 7461 (Aerospace, HW)
- NASA-GB-1740.13-96 (SW-Guidebook)
- Def Stan 00-55 (Military)
- IEC 60880 (SW in Nuclear Power Plants)

IEC 61508 Derivates
- EN 5012x (Railway)
- IEC 60601 1-4 (Medical)
- IEC 61513 (Nuclear)
- IEC 61511 (Process Industry)
- **ISO WD 26262 (Automotive)**

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche tests et validation
- Conclusion

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche tests et validation
- Conclusion

Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- Objectifs de la SdF:
  - Risque de Gravité X => comportement du système risque d'engendrer des blessures graves voire des décès
  - Risque de Gravité Y => indisponibilité gênante du système
  - Probabilités d'occurrence généralement adoptées:
    - Probabilité d'apparition d'un risque de Gravité X doit être inférieure à 10-7 / Heure de mission
    - Probabilité d'apparition d'un risque de Gravité Y doit être inférieure à 10-9 / Heure de mission

Différentes approches pour améliorer la sûreté

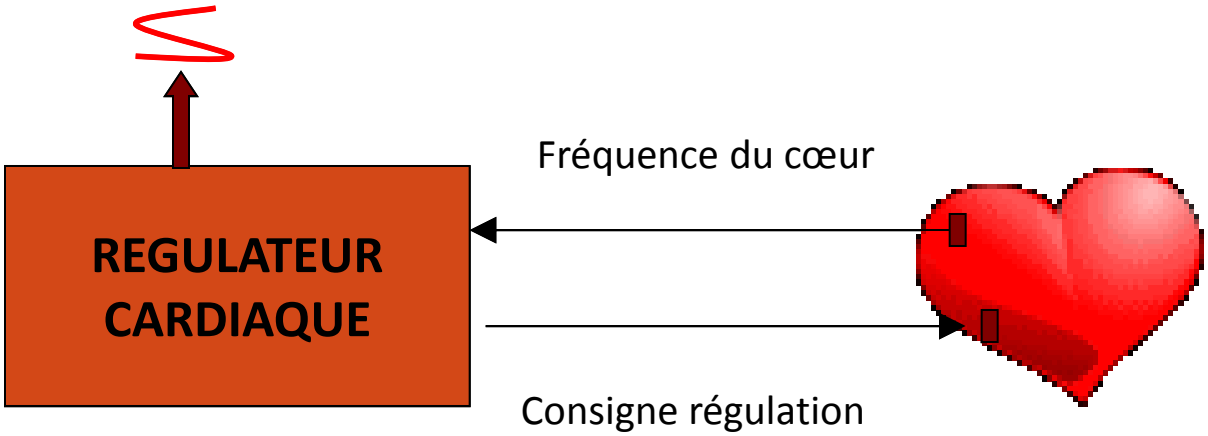
Approche sûreté de fonctionnement

- Point de départ: analyse fonctionnelle
  - Analyse préliminaire des risques (APR)
  - Analyse des modes de défaillances, de leurs effets et de leur criticité (AMDEC)
  - Analyse des effets des erreurs logicielles (AEEL)
  - Arbres de défaillances

Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- Exemple: régulateur cardiaque



Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- Exemple: régulateur cardiaque
  - Fonctionnalités et contraintes générales:
    - Performances de la régulation: 70 à 160 pulsation par minute,
  - Fonctions de services:
    - FS01: Réguler le cœur
    - FS02: Signaler par radio toute anomalie permanente
  - Fonctions de contraintes:
    - FC01: Résister à toute perturbation extérieure
  - Situations de vie:
    - Repos, Marche, Course

Exigences classiques

Différentes approches pour améliorer la fiabilité

Approche sûreté de fonctionnement

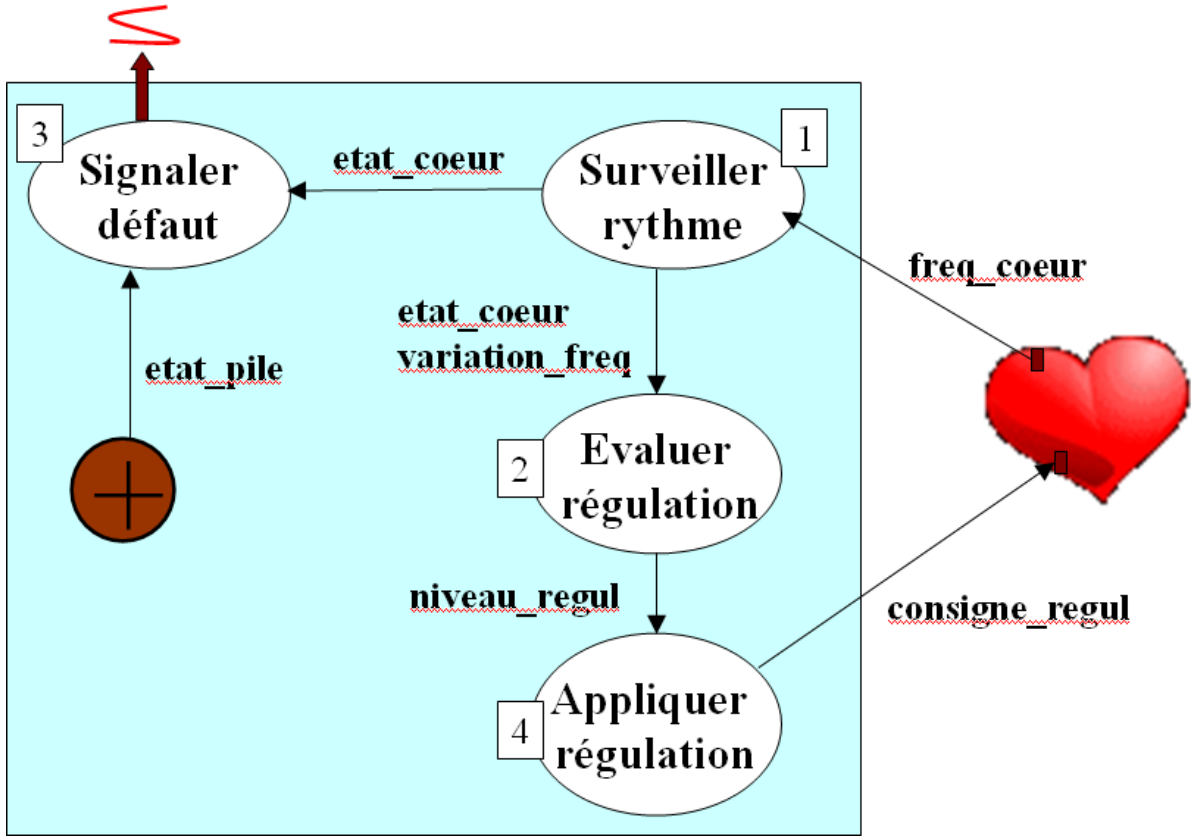
- Exemple: régulateur cardiaque
  - Critère de défaillance unique:
    - le système doit tolérer une défaillance simple logicielle ou matérielle.
  - Le MTBF (temps moyen avant la prochaine défaillance):
    - au moins trois ans
  - Exigence sûreté:
    - la probabilité d'accident grave doit être inférieure à 10–9/ h de fonctionnement
  - Définition des modes dégradés à adopter en cas de défaillance et conditions de passage entre modes.

Exigences SdF

Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- Décomposition fonctionnelle



Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- **Analyse Préliminaire des Risques**

  - L'APR est utilisée pour établir une liste, aussi exhaustive que possible, des événements redoutés d'un système afin de déterminer les risques à étudier en détail à l'aide d'autres méthodes de la Sûreté de Fonctionnement (en particulier AMDEC, Arbres de défaillance).

  - Trois étapes:

    - Rechercher et classer les **évènements redoutés** selon leur gravité

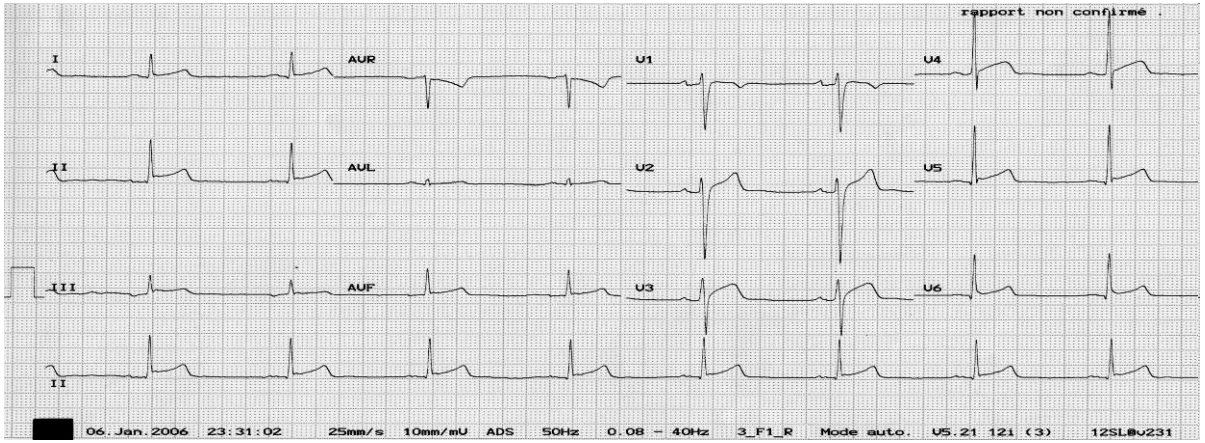
    - Identifier les scénarios générateurs de ces évènements redoutés

    - Établir des objectifs détaillés de sécurité

Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- **ER1**: mauvaise régulation du cœur (à une fréquence erronée)
- **ER2**: régulation non souhaitée du cœur
- **ER3**: absence de régulation du cœur
- **ER4**: non indication d'une situation de défaillance (permanente)
- **ER5**: fausse indication d'une situation de défaillance (permanente)



Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- APR

Evènements redoutés	Causes	Conséquences	Actions de maîtrise de risques
ER4: non indication d'une situation de défaillance	Défaillance de F3: "signaler défaut"	Non envoi du signal indiquant un défaut pile ou rythme du coeur	Redondance diversifiée de F3
	Défaillance de F1: "surveiller rythme"	Absence d'indication de défaut de rythme du coeur	Redondance diversifiée de F1 OU Surveillance de F1 + envoi d'un signal en cas de défaillance de F1
	Défaillance matérielle	---	---

Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- **Analyse des Modes de Défaillances de leurs Effets et de leurs Criticité**

  - Principe:

    - Analyse exhaustive des défaillances (une seule défaillance à le fois) pour chaque composant logiciel
    - Analyse des conséquences de ces défaillances sur le système
    - Actions correctives qui en découlent

  - Intérêt:

    - Méthode exhaustive très riche qui donne la liste de points critiques

Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- AMDEC

Fonction	Mode défaillance	Conséquences	Actions de maîtrise de risques
F1: Surveiller rythme	**Variable** *etat_coeur* **incorrecte**	Fausse Indication de défaut	Redondance diversifiée
		Absence d'indication de défaut	
	Variable *variation_freq* **incorrecte**	Mauvaise régulation du cœur	Redondance diversifiée **OU** Détection + Mode dégradé
		Absence de régulation du coeur	

- Limites de l'AMDEC:
  - Lourd à mettre en œuvre pour des systèmes complexes
  - Ne concerne que des défaillances simples
  - Les aspects séquences d'évènements n'apparaissent pas clairement

Différentes approches pour améliorer la sûreté

Approche sûreté de fonctionnement

- **Conclusion**
  - La démarche SDF commence au niveau du système complet, ce qui va ensuite décliner des exigences SDF au niveau logiciel. Il faut une cohérence entre les deux démarches.
  - La SDF logiciel ne se résume pas à une démarche de qualité logicielle, même si cette dernière est un élément essentiel pour atteindre un niveau de fiabilité élevé
  - Les logiciels sécuritaires entraînent un surcoût important en temps et en moyens techniques.
  - En plus des livrables habituels (spécification, conception, tests…) le client exige des livrables spécifiques à la SDF (APR, AMDE, modes dégradés, tests orientés SDF, …)
  - Aspects responsabilité du fournisseur en cas de défaillance entraînant des blessures graves.

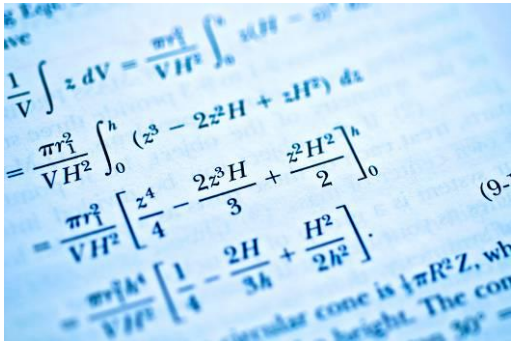
Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche
- Conclusion

Différentes approches pour améliorer la sûreté

Approche mathématique

- **Idée principale**
  - Utilisation de méthodes formelles, à fort contenu mathématique
  - Objectif : permettre de « prouver certaines propriétés » du logiciel résultant
  - Travail surtout au niveau de la spécification avec parfois un lien avec le logiciel lui-même
  - Très lié à l'idée de génération automatique de code
- **Familles**
  - Méthode B
  - Langages synchrones : Esterel, Lustre, Signal
  - Réseaux de Petri, LDS, State Charts



Différentes approches pour améliorer la sûreté

Approche mathématique

- **Conclusion**
  - Très intéressante car forte base théorique
  - Permet de prouver des propriétés, notamment de SdF (ex : Meteor)
  - Très lourd et très coûteux
    => doit être réservé aux parties les plus critiques
  - Mais il existe d'autres approches moins lourdes et aussi satisfaisantes

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche
- Conclusion / Ce qu'il faut retenir

Différentes approches pour améliorer la sûreté

Approche « qualité »

- Approche qualité **« ISO »**
  - Processus classique dans l'industrie
  - Formalisé par ISO 9000 (ISO TS 16949 pour l'automobile)
  - Bilan
    - Beaucoup de contraintes
    - Coûts élevés, sujet sensible sur certains marchés
    - Résultats assez décevants pour le logiciel
    - Peut fonctionner
      - Sous certaines conditions (approche positive et productive)
      - Une base pour s'améliorer

Différentes approches pour améliorer la sûreté

Approche « qualité »

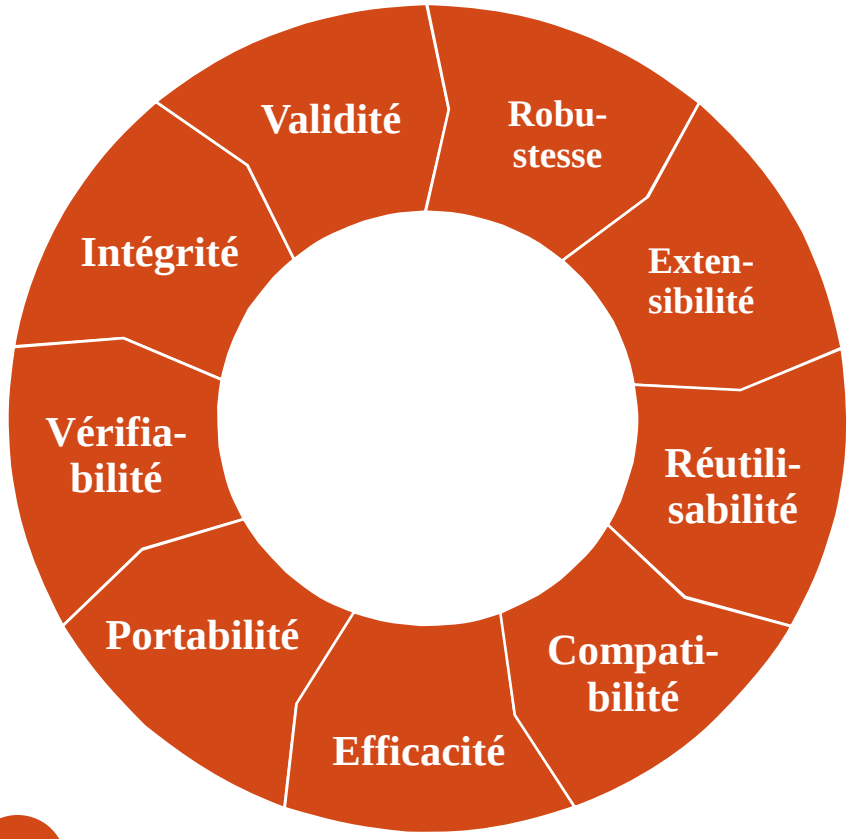
- Approche qualité **du logiciel lui-même**
  - De nombreux facteurs (= caractéristiques) permettent de quantifier la qualité d'un logiciel
  - Ces facteurs peuvent être regroupés en deux catégories:
    - Internes (visibles des équipes de développement).
    - Externes (visibles des utilisateurs).
- Un des premiers facteurs de qualité:

Le facteur minimum de qualité attendu d'un logiciel est qu'il remplisse correctement les exigences fonctionnelles qui ont conduit à son développement

Différentes approches pour améliorer la sûreté

Approche « qualité » ISO 9126 et 14598

- Facteurs de qualité internes et externes
- Evaluation des exigences qualitées



+

Facilité d'emploi

Facilité

- ▶ d'apprentissage
- ▶ d'utilisation
- ▶ de préparation des données
- ▶ d'interprétation des erreurs
- ▶ de rattrapage en cas d'erreur d'utilisation

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche tests
- Conclusion

Différentes approches pour améliorer la sûreté

Approche certification

- La conformité à une norme est très souvent imposée
- Revient à suivre les exigences de la norme concernée (voir §2 de la présente présentation)
- Peut être difficile à appréhender (surtout pour les normes orientées performances telles que la 61508 et la 26262)
- Souvent lourd à mettre en œuvre avec de fortes conséquences sur les coûts

Tableau B.2 – Analyse dynamique et test
(référencés dans les tableaux A.5 et A.9)

	Technique/Mesure*	Réf.	SIL1	SIL2	SIL3	SIL4
1	Exécution de cas de test à partir de l'analyse des valeurs aux limites	C.5.4	R	HR	HR	HR
2	Exécution de cas de test à partir de l'estimation des erreurs	C.5.5	R	R	R	R
3	Exécution de cas de test à partir de l'implantation d'erreurs	C.5.6	---	R	R	R
4	Modélisation du fonctionnement	C.5.20	R	R	R	HR
5	Classes d'équivalence et test des partitions d'entrée	C.5.7				
6	Tests basés sur la structure	C.5.8				
NOTE – L'analyse des cas de test se fait au niveau du sous-système et es spécification et le code.						
* Des techniques/mesures appropriées doivent être sélectionnées en fonctio						

	Methods and Measures					
1a	Inspection of software safety requirements					
1b	Walkthrough of software safety requirements					
1c	Model review	5.4.8	++	++	++	++
3a	Formal verification	5.4.8	+	+	++	++
3b	Semi-formal verification	5.4.8	+	+	++	++

NOTE 1: The full set of software safety requirements needs to be verified by an appropriate combination of methods 1x. This combination typically depends on the applied methods and measures for specifying software safety requirements (see table 5.1)

NOTE 2: Method 3a requires a formal specification of the software safety requirements.

NOTE 3: In method 3b, simulation of software safety requirements is regarded as a semi-formal verification. This requires a specification of the software safety requirements by executable models.

	Methods and Measures	According to req.	ASIL A	ASIL B	ASIL C	ASIL D
1	Documentation of software architectural design in natural language	6.4.1	++	++	++	++
2a	Formal methods for software architectural design	6.4.1	+	+	++	++
2b	Semiformal methods for software architectural design	6.4.1	+	++	++	++
2c	Informal methods for software architectural design	6.4.1	++	++	+	+
3	Computer-aided tools for software architectural design	6.4.1	+	+	++	++
4	Guidelines for software architectural design	6.4.1	+	++	++	++

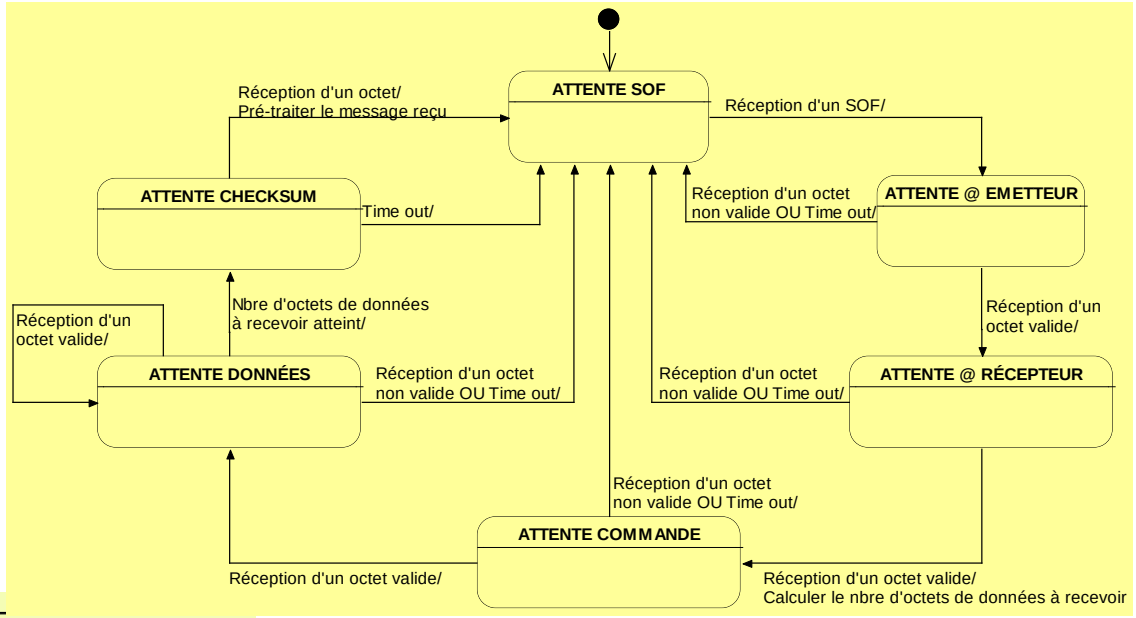
Agenda

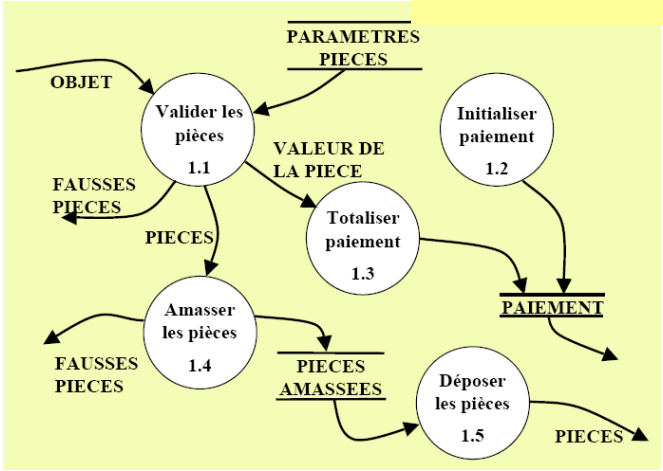
- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche tests et validation
- Conclusion

Différentes approches pour améliorer la sûreté

Approche méthodologique

- Approche méthodologique
  - UML
  - SA/RT





Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche tests et validation
- Conclusion

Différentes approches pour améliorer la sûreté

Approche « Process »

- Voir cours CMMI et SPICE

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche tests et validation
- Conclusion

Différentes approches pour améliorer la sûreté

Approche Tests & Validation

- Définir une validation poussée du logiciel
- Pas uniquement le fonctionnement nominal
- Surtout tester la robustesse du logiciel à toutes les perturbations externes

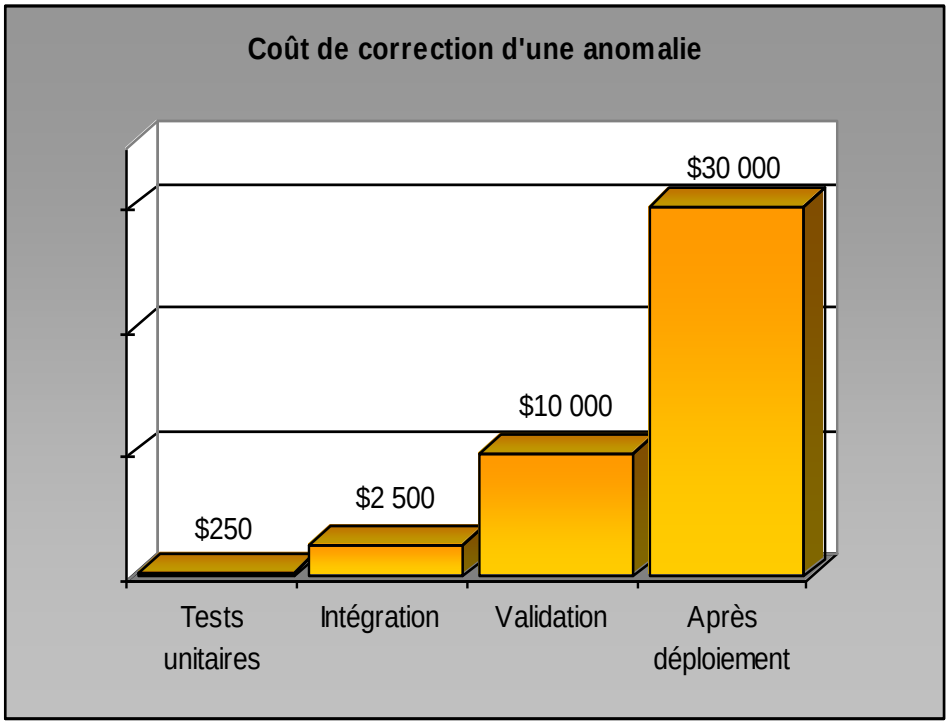
« Les méthodes de tests de logiciels traditionnelles sont inefficaces et inopérantes. La plupart des organisations dépensent, sans le savoir, la moitié de leurs ressources de développement à faire des tests… lesquels ne testent que 50 % du code. »

Gartner group

Différentes approches pour améliorer la sûreté

Approche Tests & Validation

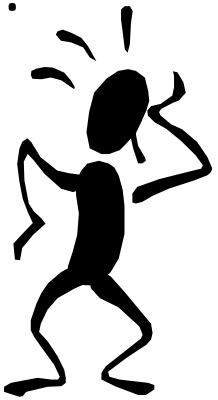
- **Constats**
  - Pas (ou peu) de méthodologie pour guider la démarche de validation.
  - Intégration souvent difficile, et phase de validation longue et parfois insuffisante.
  - Problèmes de qualité du logiciel.
  - Non-respect des plannings, et des budgets.
  - Rappels produits
  - Retards de mise sur le marché et coûts induits



Différentes approches pour améliorer la sûreté

Approche Tests & Validation

- Trop souvent les tests commencent après l'intégration.
  - Test de la partie "visible" de l'applicatif uniquement,
  - Risque de découvrir (trop tard) des problèmes de fond
- Méthode proposée
  - Intégrer le test dans le processus de développement, en adoptant une démarche structurée.
    - Préparation du processus de validation lors des phases de développement.
    - Utilisation de maquettes du logiciel.



Tester et intégrer au fur et à mesure

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté
  - Approche sûreté de fonctionnement
  - Approche « mathématique »
  - Approches « qualité »
  - Approche certification
  - Approche méthodologique
  - Approche « process »
  - Approche tests et validation
- Conclusion

Différentes approches pour améliorer la sûreté

Quelle est la meilleure approche?

- Hélas, pas de « vérité vraie »
- Elles sont toutes bonnes…car elles ne sont pas contradictoires
- Plutôt les combiner pour obtenir le résultat attendu

Différentes approches pour améliorer la sûreté

Quelle est la meilleure approche?

- Attention à ne pas utiliser « un marteau-pilon pour écraser une mouche ! »
- Il faut les utiliser intelligemment
- Attention:
  - à ne pas les appliquer toutes de A à Z
    - Prendre à tout moment ce qu'il y a de bon dans chacune d'elles…
  - à ne pas produire trop de papier…

Différentes approches pour améliorer la sûreté

Quelle est la meilleure approche?

- Le bon usage de chaque approche:
  - La SdF
    - Excellente démarche d'analyse
    - Propose des solutions techniques pour améliorer la conception
    - Approche assez lourde à réserver aux cas critiques
  - Les maths – les specs formelles
    - Excellente démarche de spécification non ambiguë
    - Souvent également assez lourde
    - Possibilité d'exécuter la spec et de la valider
    - Plus intéressant que la génération automatique de code
  - Qualité
    - Plutôt la résultante de toutes les autres approches
    - Peut devenir rapidement « bestiale »
      - On ne réfléchit plus, on applique un modèle…

Différentes approches pour améliorer la sûreté
Quelle est la meilleure approche?

- Le bon usage de chaque approche:
  - Certification
    - Faire d'une obligation une opportunité d'améliorer le processus d'étude
  - Les méthodes
    - Permet de fiabiliser la démarche de spécification et de conception
    - Attention à ne pas en faire de trop
      - Le but n'est pas la spec mais le logiciel…
  - Les process
    - Proche de la qualité mais plus orienté « métier »
    - Indispensable dès que la société et / ou les projets deviennent conséquents
    - Attention à la mise en œuvre (au besoin, se faire aider par des experts)
  - Les tests et la validation
    - Finalement ce dont on parle le moins alors que c'est la seule manière de vérifier le résultat de toutes les autres approches

Agenda

- Introduction au SDF
- Rappel de la problématique pour le logiciel
- Tour d'horizon des différentes normes s'appliquant au logiciel
- Différentes approches pour améliorer la sûreté du logiciel
- Conclusion

Conclusion

- Objectifs difficiles à atteindre car les sujets sont complexes et multiformes
- Prendre conscience de l'ampleur de la tâche
- Avoir une vraie démarche d'ingénierie pour maîtriser le cycle de développement
- Combiner les différentes approches présentées en fonction des caractéristiques du projet
- Accepter de payer le vrai coût du développement au lieu de payer « les pots cassés »

Merci de votre attention…

Avez-vous des questions?

Contact:
ali.baktash@smile.fr
ali.baktash@autoliv.com

Présentation inspirée de documents rédigés par:
Christophe Brunschweiler, Smile
Bruno Compin, RENAULT