SÛRETÉ DE FONCTIONNEMENT

« OU COMMENT S'ASSURER QU'UN SYSTÈME EST SÛR ? »

B. Monsuez
ENSTA PT

DÉVELOPPEMENT D'UNE APPLICATION CRITIQUE

Intérêt d'une méthodologie

- Importance du processus pour développer et maintenir une application
  - Compromis à trouver entre fonctionnalité, performance & qualité de service
- Importance de la maîtrise des coûts
  - Nécessité d'optimiser chacune des phases de développement
  - Nécessité de limiter les phases de redesign
  - Nécessité de capturer les fautes (erreur de conception) dès leur apparition

La complexité

La complexité interne (Dvorak, 2009)

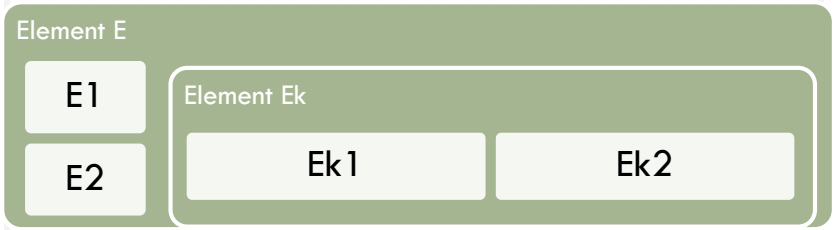
- La complexité des fonctions à réaliser
- L'environnement d'exploitation
- La criticité

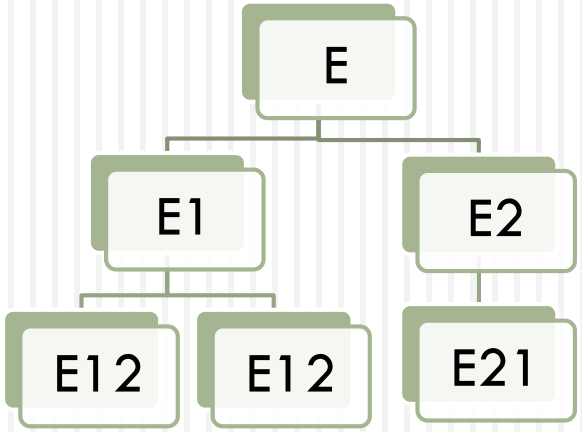
La complexité externe

- La nouveauté
- Le temps de développement
- L'équipe
- La maturité organisationnelle
- Les outils

Gérer la complexité

- Décomposition hiérarchique





- Quid des « nœuds »
  - Soit des fonctions « function trees »
  - Soit des composants « product trees »
  - Soit une combinaison des deux.

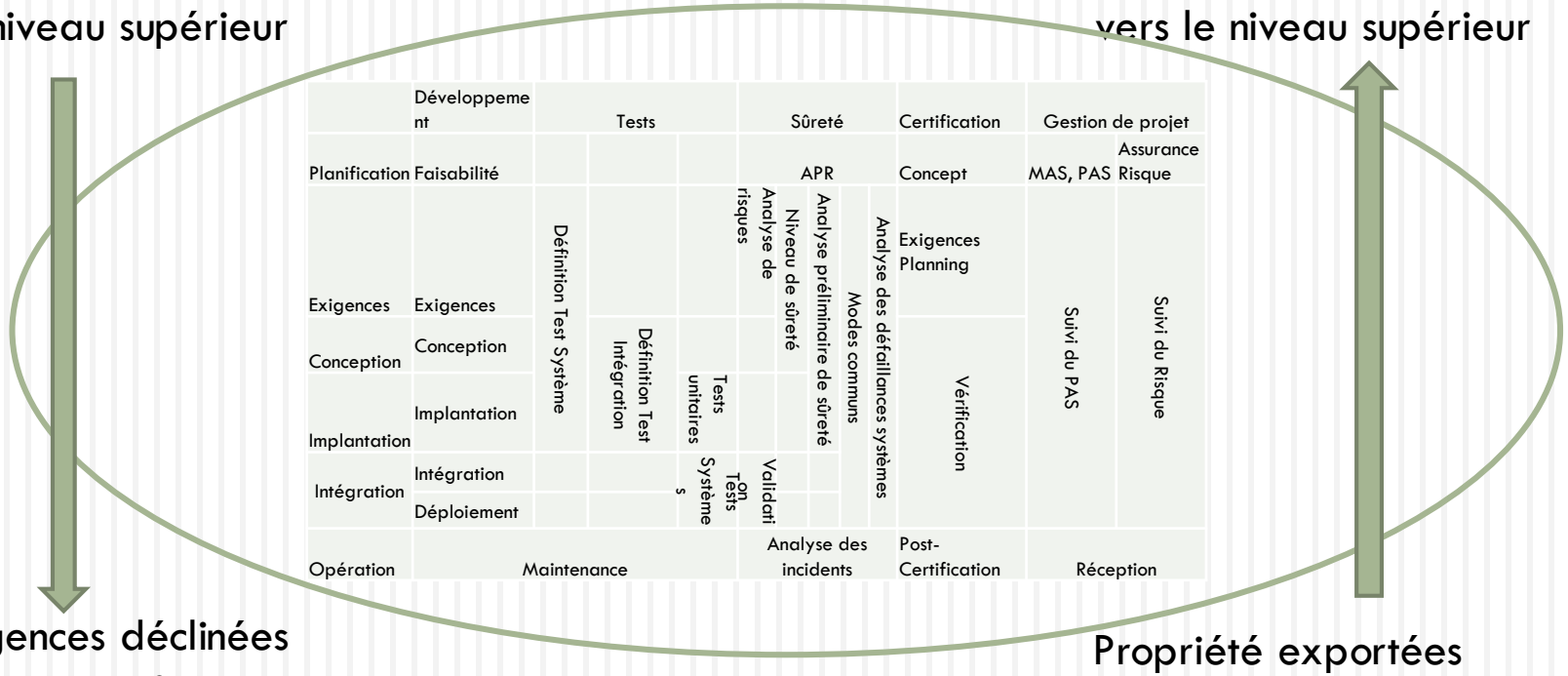
Gérer un programme complet

	Développement	Tests			Sûreté					Certification	Gestion de projet	
Planification	Faisabilité				APR					Concept	MAS, PAS	Assurance Risque
Exigences	Exigences	Définition Test Système	Définition Test Intégration		Analyse de risques	Niveau de sûreté	Analyse préliminaire de sûreté	Modes communs	Analyse des défaillances systèmes	Exigences Planning	Suivi du PAS	Suivi du Risque
Conception	Conception			Tests unitaires						Vérification		
Implantation	Implantation											
Intégration	Intégration			Tests Systèmes	Validation							
	Déploiement											
Opération	Maintenance				Analyse des incidents					Post-Certification	Réception	

Prise en comptes de la sûreté

Exigences
du niveau supérieur

Propriété exportées
vers le niveau supérieur



Exigences déclinées
au niveau inférieur

Propriété exportées
pour chacun des composants
au niveau supérieur

Analyse des risques

APR ou PHA

- Réalisée quand l'information initiale est disponible

AR ou HA

- Réalisée quand une information détaillée et plus précise est disponible durant la conception.

Analyse des risques

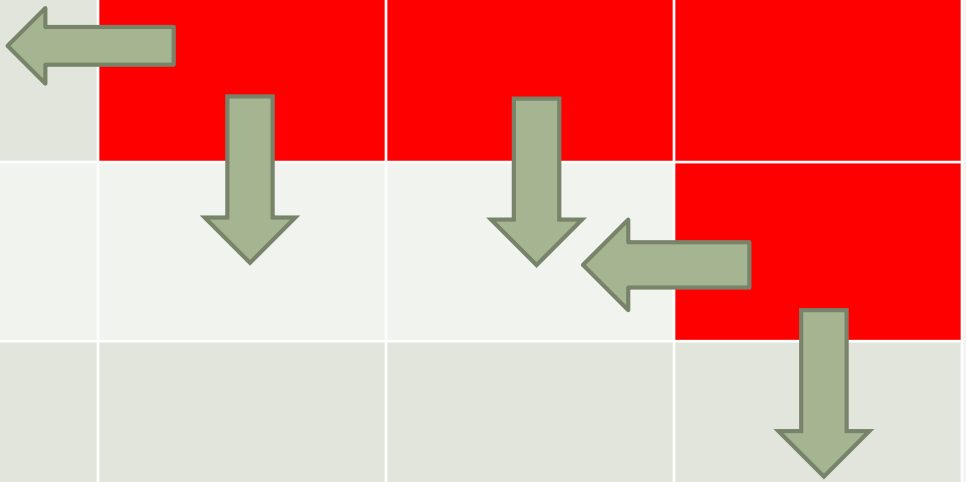
- Consiste à déterminer les évènements redoutés
  - Checklist
  - Réflexion/Expérience
  - Scenarii Si Alors
  - Contraintes normatives
  - Standards d'analyse
- Déterminer les conséquences de ces évènements redoutés
- Priorisation des évènements en fonction de leur gravité et de leur probabilité de survenance.

Détermination du niveau de sûreté requis

- Détermination des conséquences du dysfonctionnement d'un composant ou sous-système
  - **Catastrophique** : conséquence potentiellement dangereuses
  - **Dangereux** : mis à mal de la sûreté d'exploitation (violation d'une ou plusieurs règles), peut avoir des conséquences sur les occupants.
  - **Majeur** : réduction importante des marges de sûreté, charge importante des opérateurs.
  - **Mineur** : réduction des marges de sûreté, charge plus importante pour les opérateurs et inconvénients pour les occupants.
  - **Négligeable** : aucun effet sur la sûreté.

La matrice de probabilité

	Négligeable	Mineur	Majeur	Dangereux	Catastrophique
Probable ($>10^{-5}$)					
Improbable ($<10^{-5}$)					
Très Improbable ($<10^{-9}$)					



La gestion du risque

- 3 politiques de gestion du risque
  - Eliminer le risque
  - Réduire le risque : ie. réduire la probabilité d'occurrence
  - Transférer le risque : faire supporter les conséquences par un assureur.

Vérification du système

- Modèle abstrait
  - Définition initiale de l'architecture
  - Prise en compte des exigences
  - Exploration des solutions
  - Allocation des composants
  - Estimations préliminaires du niveau de sûreté et de performances

Evaluation préliminaire

- Evaluation par scénarii
  - Limitée à quelques scénarii
  - Injection de fautes indépendantes
  - Pas de couverture totale
- Evaluation par des méthodes formelles
  - Démonstration de certaines propriétés (ie. propriétés d'atteignabilité)
  - Nécessite de procéder à des abstractions

Notion d'abstraction du système

- Preuve d'une propriété sur une abstraction du système (3 cas de figure)
  - Les résultats dans le modèle abstrait sont aussi valides dans le domaine concret.
  - Les résultats dans le modèle abstrait sont valides si et seulement si ils sont valides dans le modèles concret.
  - Les résultats dans le modèle concret sont valides dans le domaine abstrait.
- Problème : parfois nous ne couvrons pas tous les cas.

Définition du flot de test

- Tests de développement
  - Ensemble des tests effectués durant le développement
- Tests de vérification
  - Tests effectués en fin de développement pour vérifier la validité du produit
- Tests de production
  - Tests effectués durant la production pour vérifier que la qualité du produit

Les phases de tests

- La planification
  - Définition des plans et des objectifs généraux de testabilité.
- La définition
  - Définition des stratégies de tests et des scénarii
- L'exécution
  - La réalisation de l'ensemble des tests.

Définition des tests de qualification

- L'ensemble des tests permettant de garantir la conformité du produit avec les exigences
  - Détermine l'ensemble des propriétés devant être vérifiées de manière globales
  - Détermine l'ensemble des propriétés devant être vérifiées lors de l'intégration
  - Détermine l'ensemble des propriétés devant être vérifiées lors des test unitaires

Définition des tests d'intégration

- Spécification de l'ensemble des tests permettant de valider l'interaction des modules

- Vérification
  - Du fonctionnement
  - Du respect des performances
  - De la robustesse

Définition des tests unitaires

- Vérification du composant
  - Relecture de l'implantation du composant (code, document de réalisation, document de design,…)
  - Définition des tests devant être réalisées pour chacun des composants
    - Définition des vecteurs de tests
    - Utilisation de méthodes formelles (Model Checking Preuves, Langages Synchrones, …)