

# INGÉNIERIE DU SYSTÈME

N. MACHKOUR

30/03/2020

N.MACHKOUR



Département GE  
2ème année CI : MSEI, GEM

2019-2020

1

# PLAN DU COURS

- Chapitre 1 : Définitions et concepts fondamentaux
- Chapitre 2 : Méthodes et processus de l'IS
- Chapitre 3 : Modélisation des exigences
- Chapitre 4 : MBSE, SysML et Outils de modélisation

# **CHAPITRE 1 : DÉFINITIONS ET CONCEPTS FONDAMENTAUX**

# PLAN

- Définitions : Systèmes, Système exigeant l'IS, IS ..
- Structure des systèmes : vue externe, vue interne
- Cycle de vie d'un système
- Approches de l'IS

# Définitions

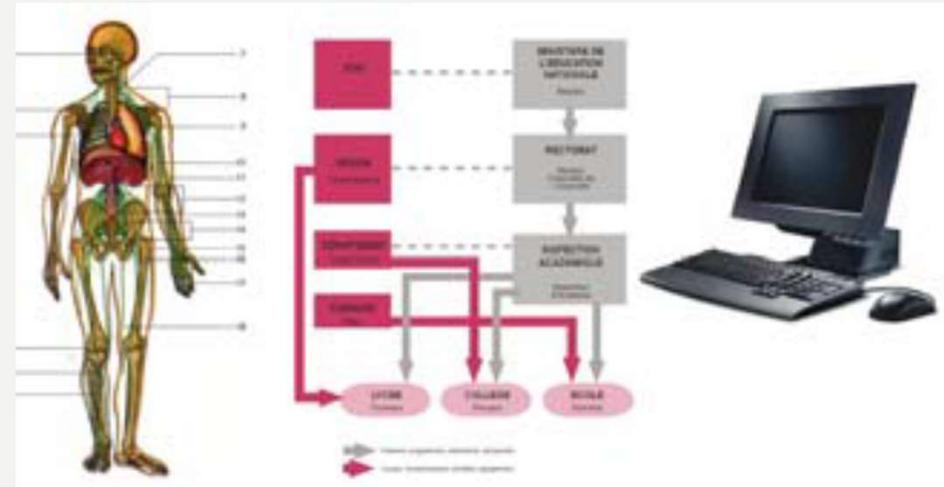
# SYSTÈME

## Définition

Un système est **un assemblage**, une collection organisée (possédant une structure) **d'objets reliés** (en interrelation) les uns aux autres, de façon à former une entité (un tout) et en relation avec le milieu extérieur dans le but de **remplir une ou plusieurs fonctions**.

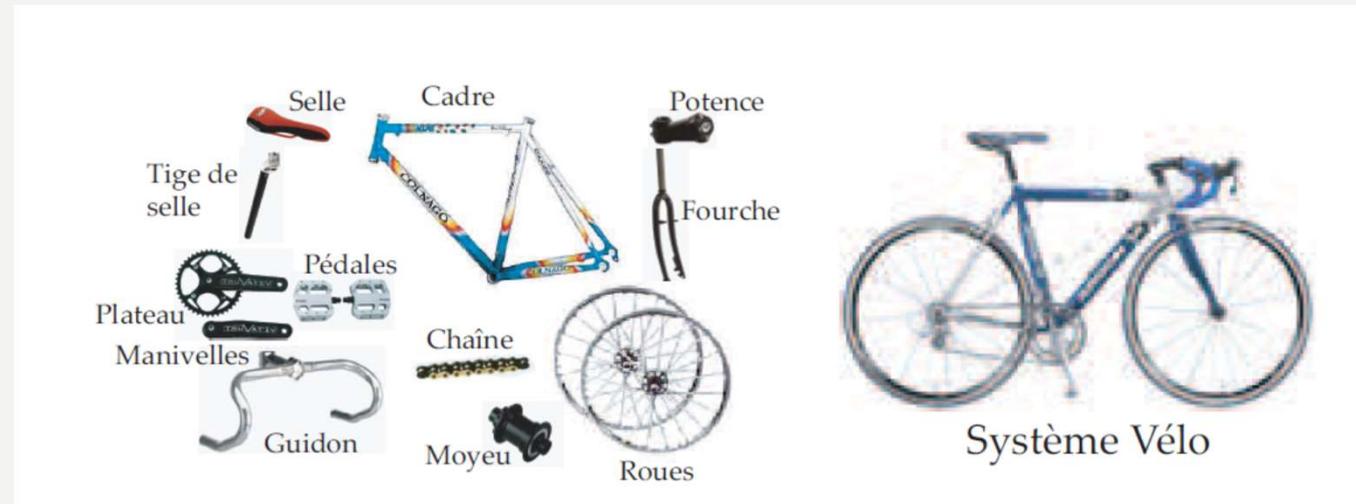
## Exemples

- Systèmes nerveux
- Systèmes informatiques
- Systèmes industriels
- Système d'équations



## Remarque

Un système n'est pas un ensemble Il suffit de connaître tous les éléments d'un ensemble pour connaître l'ensemble. Mais il ne suffit pas de connaître tous les composants d'un système pour connaître le système, il faut aussi connaître les relations entre les composants



## Complexité des systèmes

- **Un système complexe** est un système qui contient plusieurs éléments différents et combinés d'une manière qui n'est pas immédiatement claire pour l'esprit, qui est difficile à analyser.
- **un système simple** c'est un système où une information d'entrée permet de trouver le comportement de sortie par une analyse déductive.

## Exemples

- Un système industriel est **complexe** puisque les fonctions résultent de l'ensemble des interactions entre les différents composants sont difficile à analyser.
- Un système d'équations **est simple**, il nécessite une **démarche deductive** pour le résoudre une équation différentielle

- **Les systèmes simples**

- ✓ sont traités en **Sciences Fondamentales** et font appel à l'esprit de logique.
- ✓ Les problèmes sont bien posés et il y a souvent une solution unique.
- ✓ Leur comportement se place dans le cadre des modèles de connaissances : le comportement de chaque composant est géré par un phénomène physique élémentaire, et le comportement du système entier est la somme des comportements élémentaires

### Les systèmes complexes

- ✓ leur analyse est réalisée en **Sciences Industrielles** pour l'Ingénieur.
- ✓ nécessite de modéliser le comportement des composants du système et de leurs interactions.
- ✓ sollicite l'esprit de synthèse et d'initiative.

Système complexe



Son comportement global n'est donc pas directement prévisible à partir des comportements élémentaires des composants

## ORIGINE DE L'INGÉNIERIE DE SYSTÈME

- ***Des constats***

Echecs technologiques, dépassement de délai et de coût, pertes humaines, rejets sociaux, conséquences écologiques, ...

- ***Des causes***

- ✓ Complexité grandissante des systèmes (pluridisciplinaires voire interdisciplinaires)
- ✓ Manque de vision globale pour l'ingénierie et le management
- ✓ Ensembles réalisés par juxtaposition de technologies
- ✓ Interfaces techniques défectueuses
- ✓ Difficultés à faire dialoguer les métiers et les organisations
- ✓ Peu de "spécialistes" pluridisciplines

- ***Une réponse***

Maîtriser les systèmes complexes par une ingénierie structurée

## HISTOIRE DE L'IS

- les premiers organismes à s'intéresser à l'IS ont été les grandes institutions américaines de la défense :
  - ✓ La NASA (National Aeronautics and Space Administration) et l'USAF (United States Air Force) ont tenté, dans les années 1960, de cadrer le développement des programmes militaires et d'exploration spatiale, au travers d'approches industrielles plus rationnelles.
- en 1991 : création de l'International Council On Systems Engineering INCOSE : premier organisme mondial d'IS, tant par sa date de création que par sa taille.
- formalisée de manière rigoureuse à partir de 2001 par les normes IEEE 1220, EIA 632 et ISO 15288.
- Actuellement, l'ingénierie des systèmes est maintenant reconnu comme une profession

# L'IS QU'EST CE QUE C'EST ?

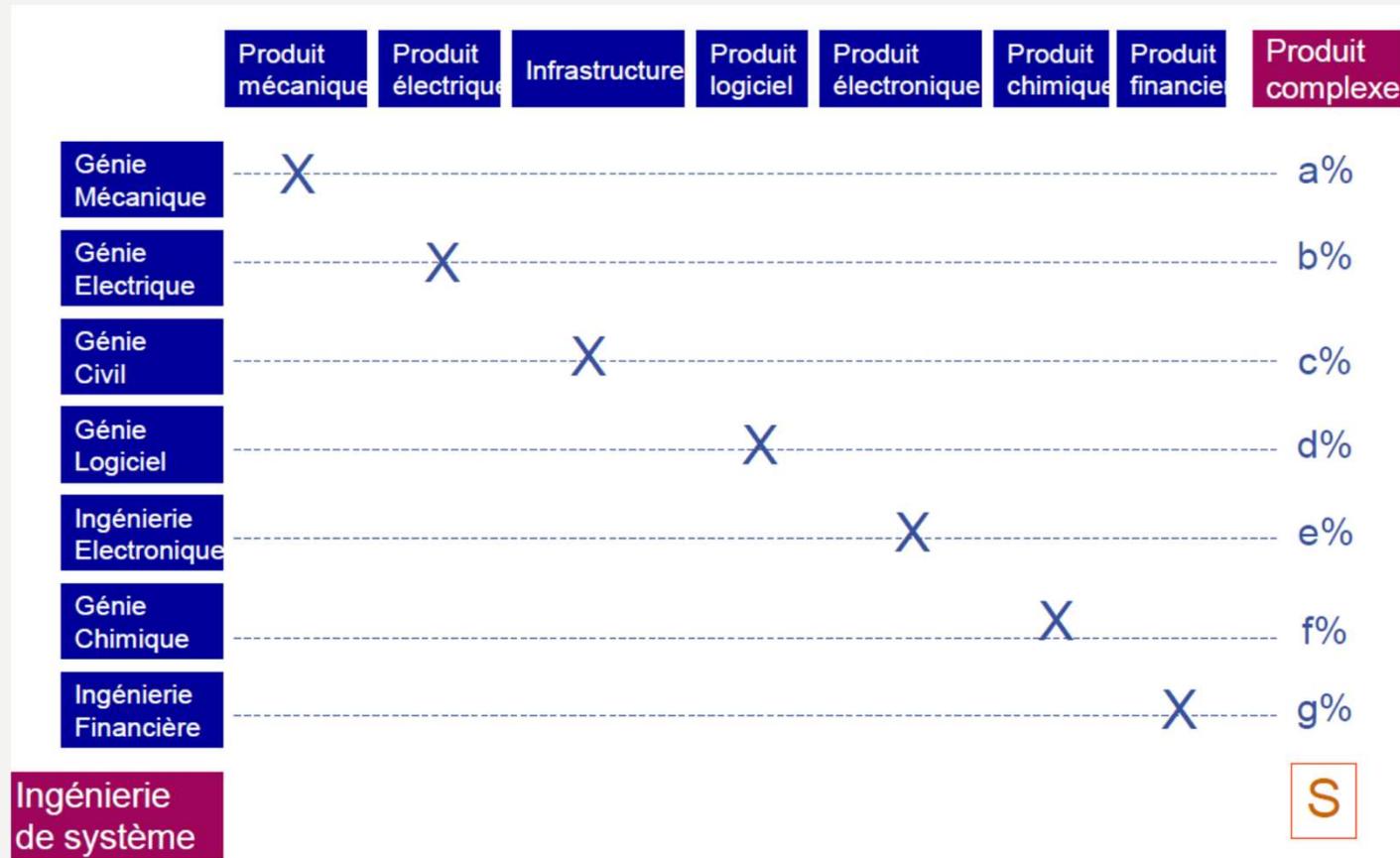
- selon **L'INCOSE**, *l'IS* est définie comme une approche interdisciplinaire et des moyens qui permettent la réalisation des systèmes réussis.
- selon **L'AFIS 2009**, *l'IS* est une approche globale et systématique apte à mettre en cohérence les méthodes et les démarches existantes, tel que l'analyse fonctionnelle, l'analyse de la valeur, la sûreté de fonctionnement, la conception à objectifs désignés, les méthodes d'ingénierie de spécialités ou de génie des métiers impliqués, etc., tout en bénéficiant de leurs retours d'expériences
- *Ingénierie Système* est définie comme un mode de pensée et une façon d'appréhender les affaires par une approche structurée pour passer du besoin à la solution.

# SYSTÈME EXIGEANT LA PRATIQUE DE L'IS

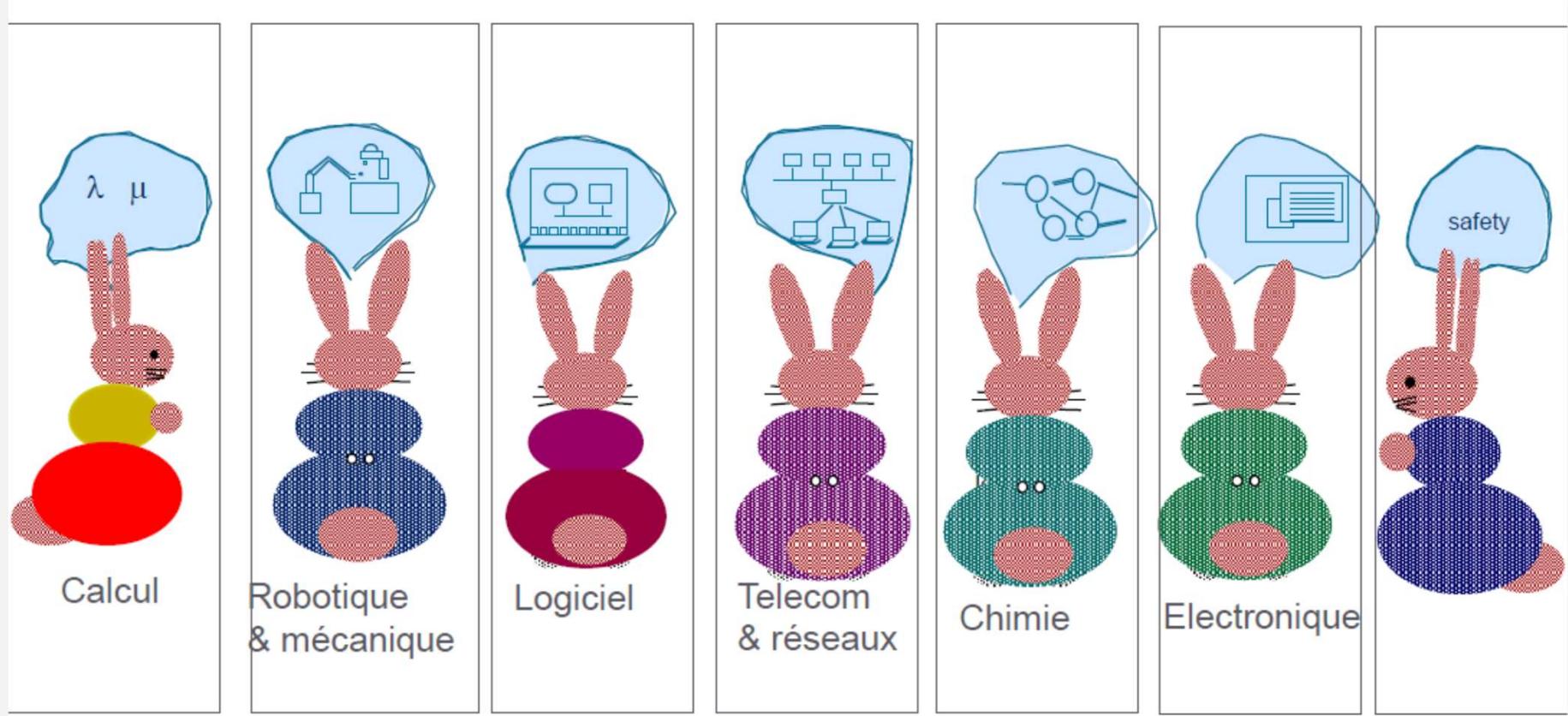
Les caractéristiques d'un système dont le développement, les essais et l'application exigent la pratique de l'ingénierie des systèmes sont le système qui :

- Est **un produit d'ingénierie** et répond donc à un besoin spécifique ;
- Se **compose de diverses composantes** qui ont des **relations complexes** entre elles et qui sont donc multidisciplinaires et relativement complexes ;
- **Utilise des technologies de pointe** d'une manière qui est essentielle à l'exécution de **ses fonctions principales** et qui comporte donc un risque de développement et, souvent, un coût relativement élevé.

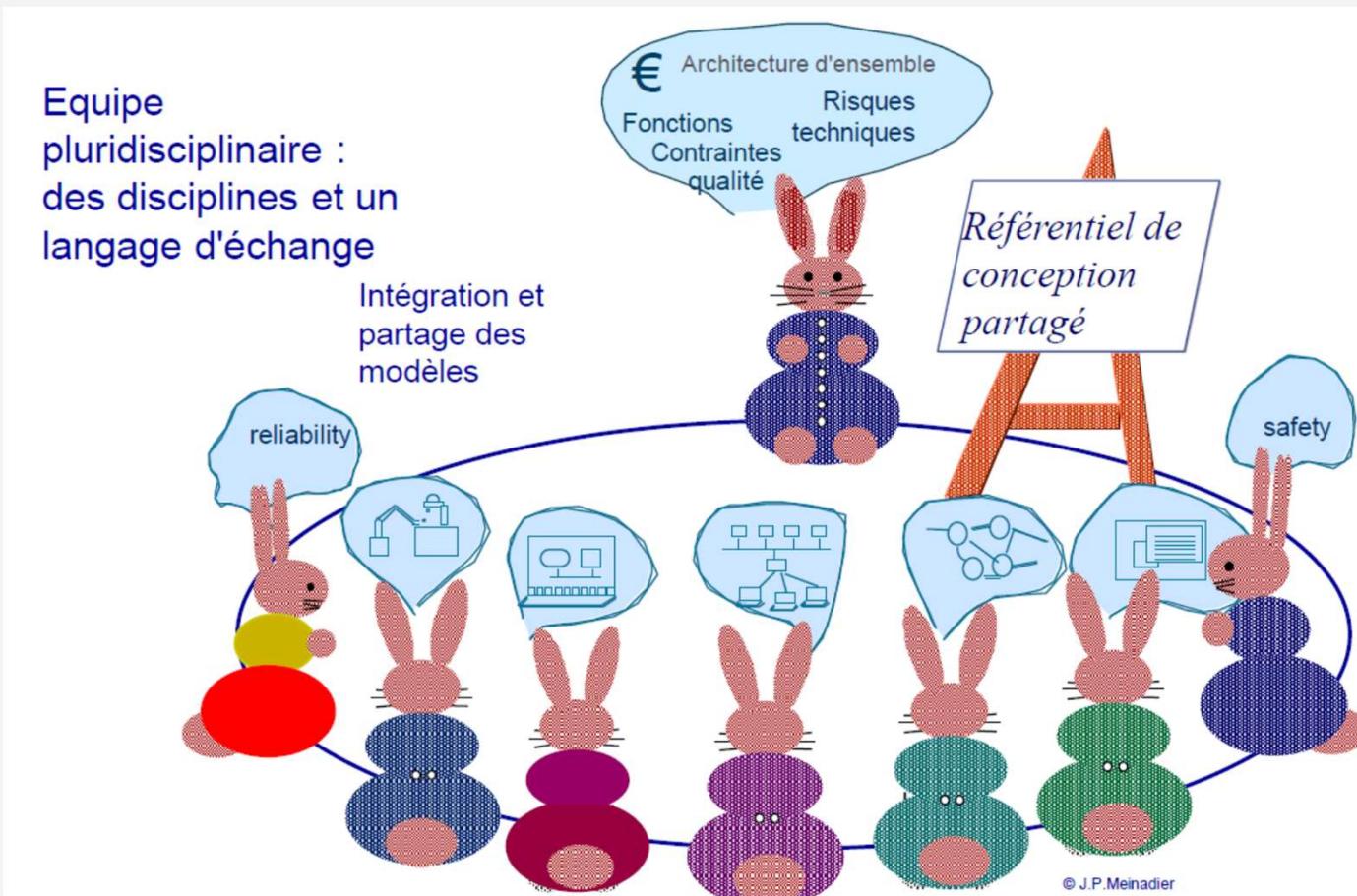
# DE LA MONO-TECHNOLOGIE À LA PLURIDISCIPLINARITÉ



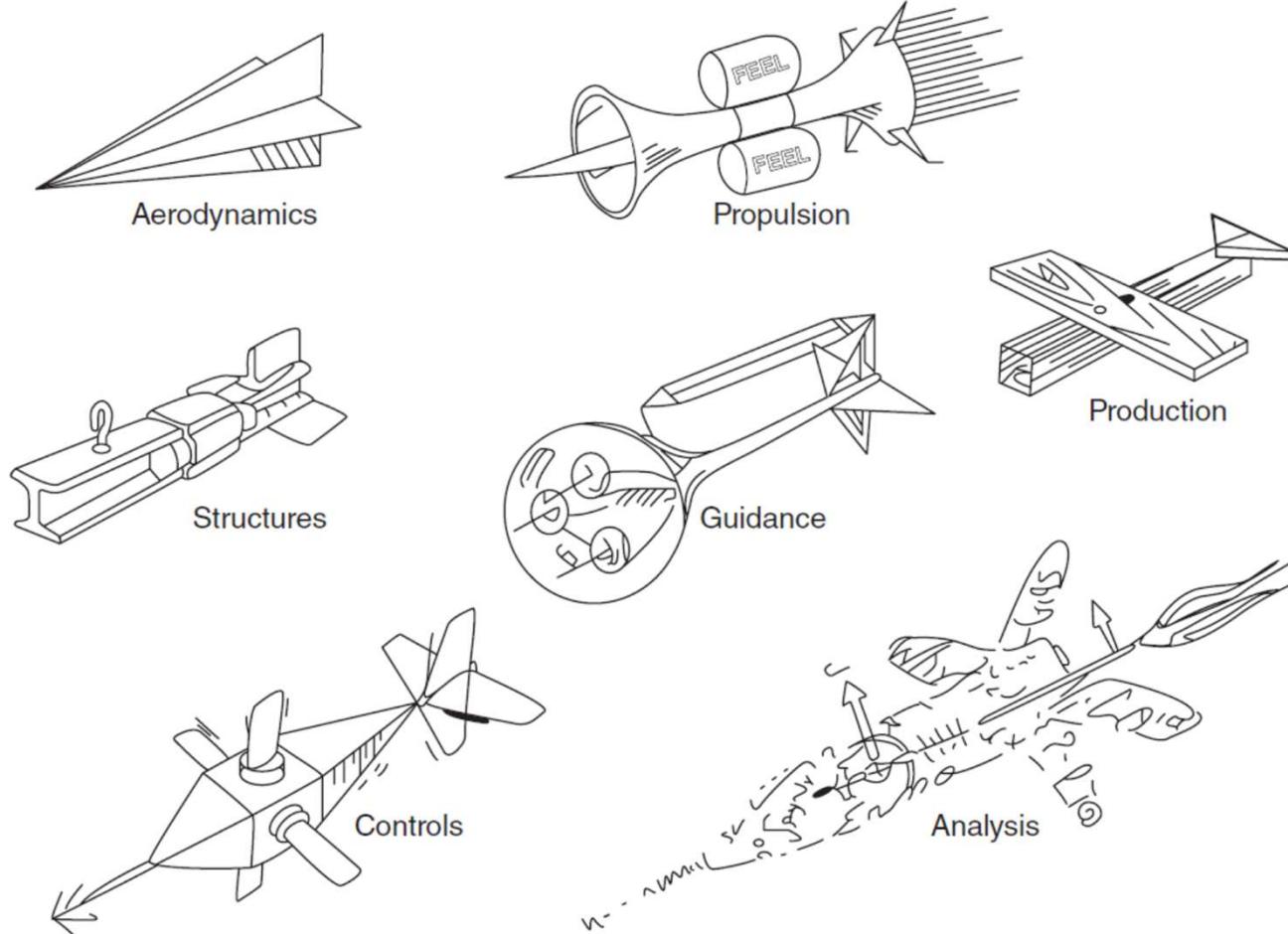
# L'IS : INTERDISCIPLINARITÉ ET PARTAGE DES MODÈLES



# Interdisciplinarité et partage des modèles

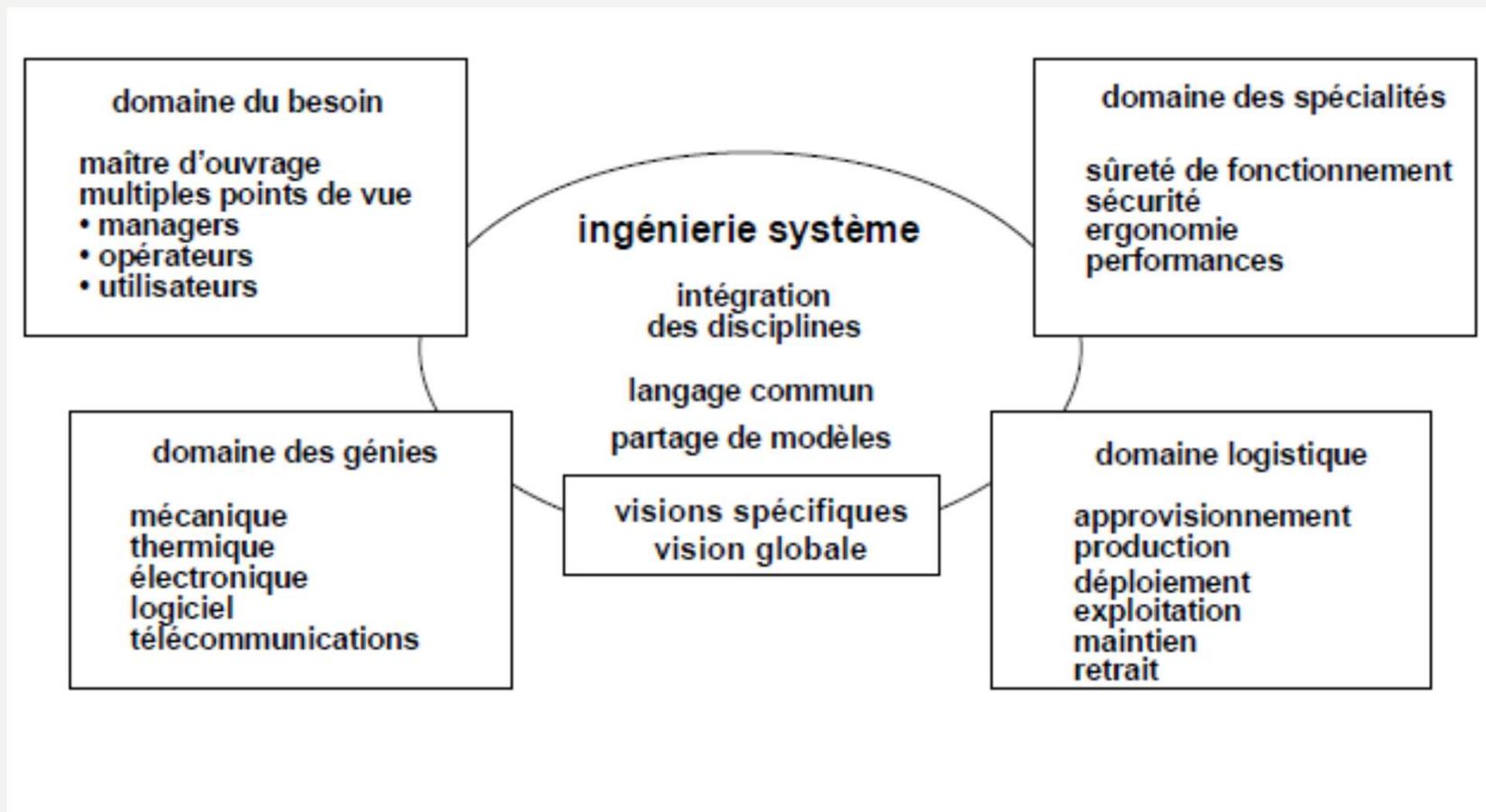


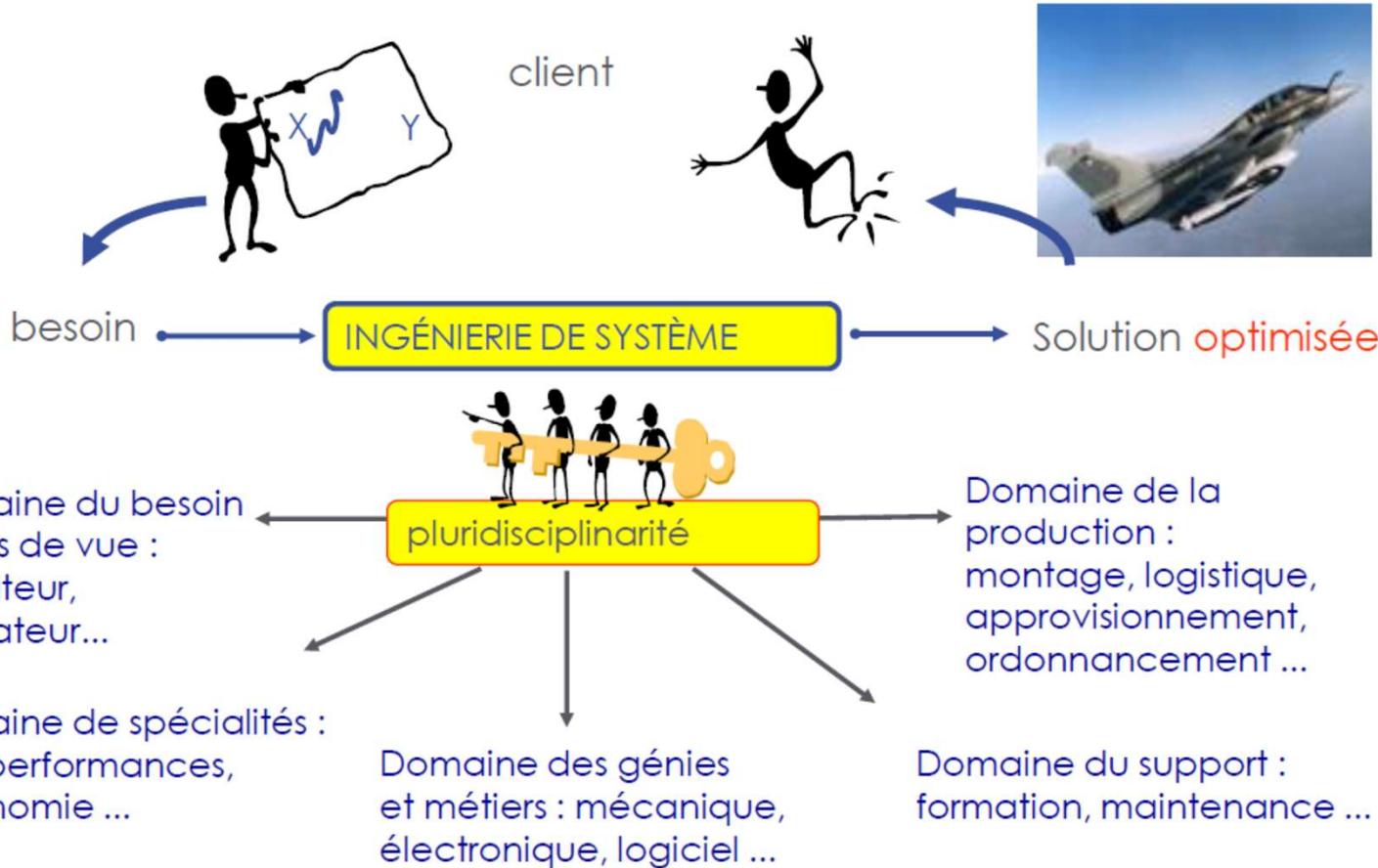
## Exemple



Il s'agit d'une conception artistique de ce à quoi pourrait ressembler un missile guidé s'il était conçu par un spécialiste de l'un des composants de ce missile

# EN CONCLUSION L'IS : PLURI ET INTERDISCIPLINARITÉ

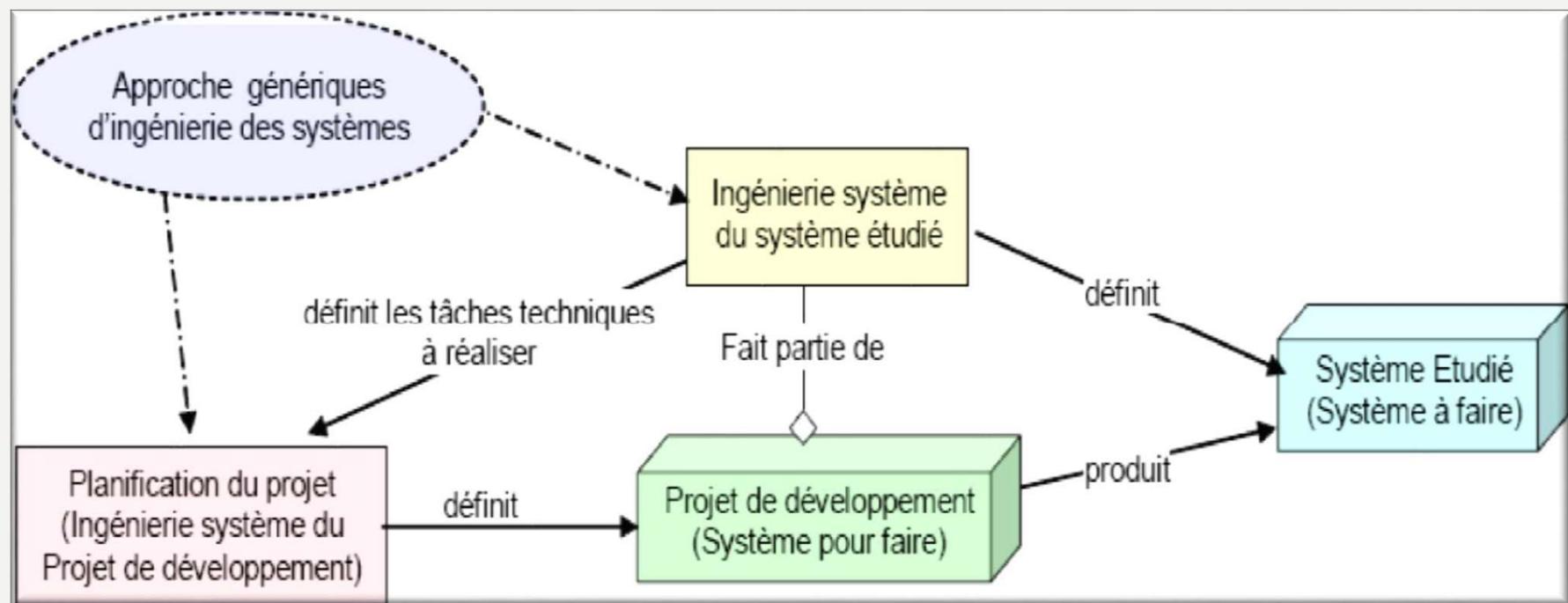




## *Une méthode de résolution de problèmes complexes*

# INGÉNIERIE SYSTÈME -SYSTÈME PROJET

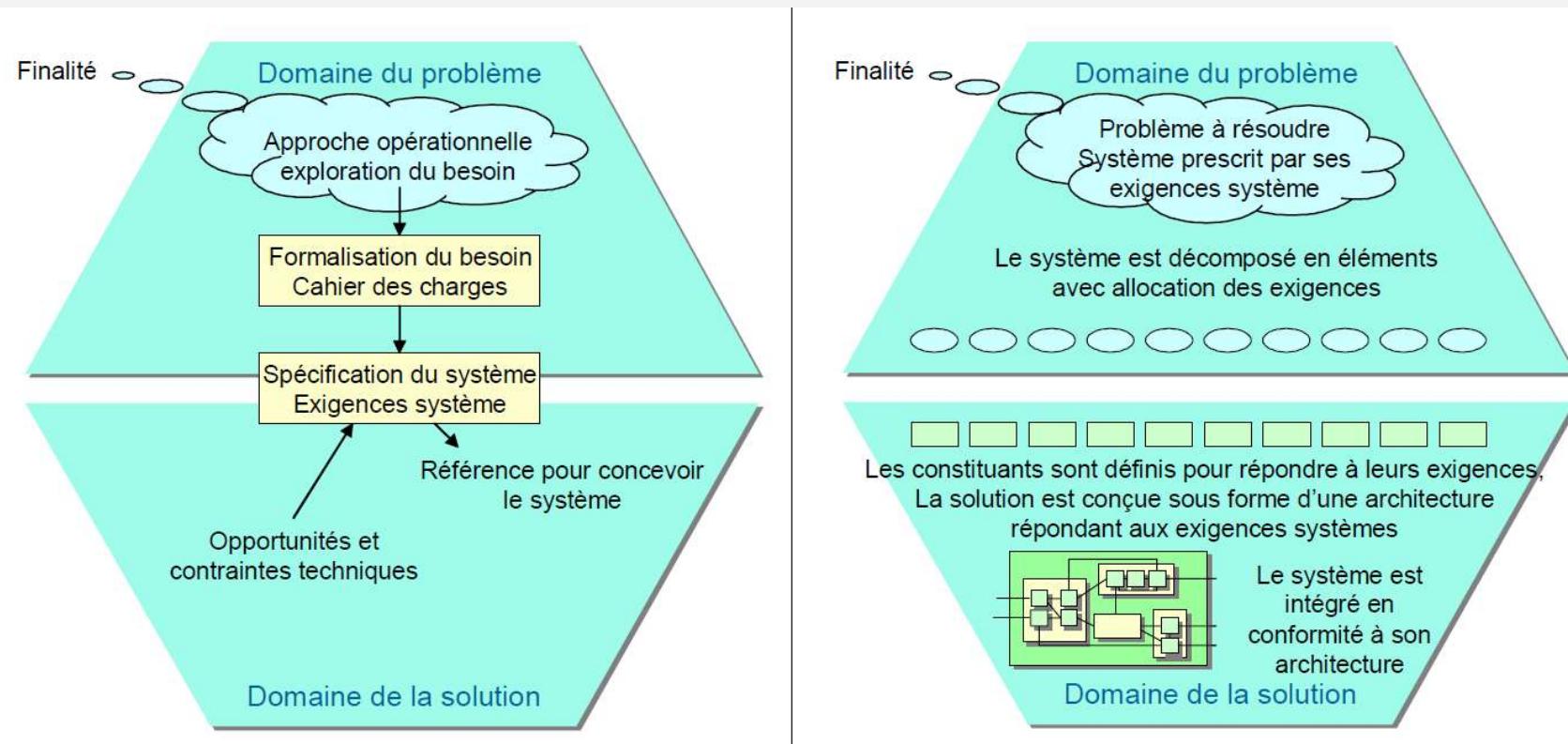
- Des méthodes pour l'ingénierie du :
  - ✓ Système étudié : système à faire
  - ✓ Système projet : système pour faire



# Structure des systèmes complexes

Deux vues complémentaires :

- Vue externe : le système est une « boite noire»
- Vue interne : le système est une « boite blanche »



## VUE EXTERNE :

*« le système est en interaction dans son environnement »*

- Elle considère l'aspect externe uniquement, en ignorant la constitution du système pour ne considérer que ses entrées et sorties et les effets de son action sur l'environnement,
- C'est la représentation du système sans considérer son fonctionnement interne,
- Elle (focus maîtrise d'ouvrage) se préoccupe des échanges et interfaces entre constituants de l'écosystème. La modélisation des comportements pour cela est un outil précieux.

## VUE INTERNE :

*« le système a une architecture fonctionnelle et organique »*

- Le système est une boite blanche (vision de maître d'œuvre),
- Le système est décomposé en sous système par décomposition itérative en élément fonctionnels et techniques jusqu'on obtienne des éléments suffisamment simples pour pouvoir les acquérir ou les réaliser,
- les exigences sont raffinées sur les éléments de décomposition,
- les élément sont intégrés selon l'architecture du système vue sur le plan fonctionnel (ce qui fait le système) et sur le plan organique (ce qui est le système ),
- le système obtenu est vérifié si il est conforme aux exigences système et si il se comporte conformément aux scénarios opérationnels spécifiés dans l'environnement spécifié.

# Vue interne : description logique et physique des systèmes

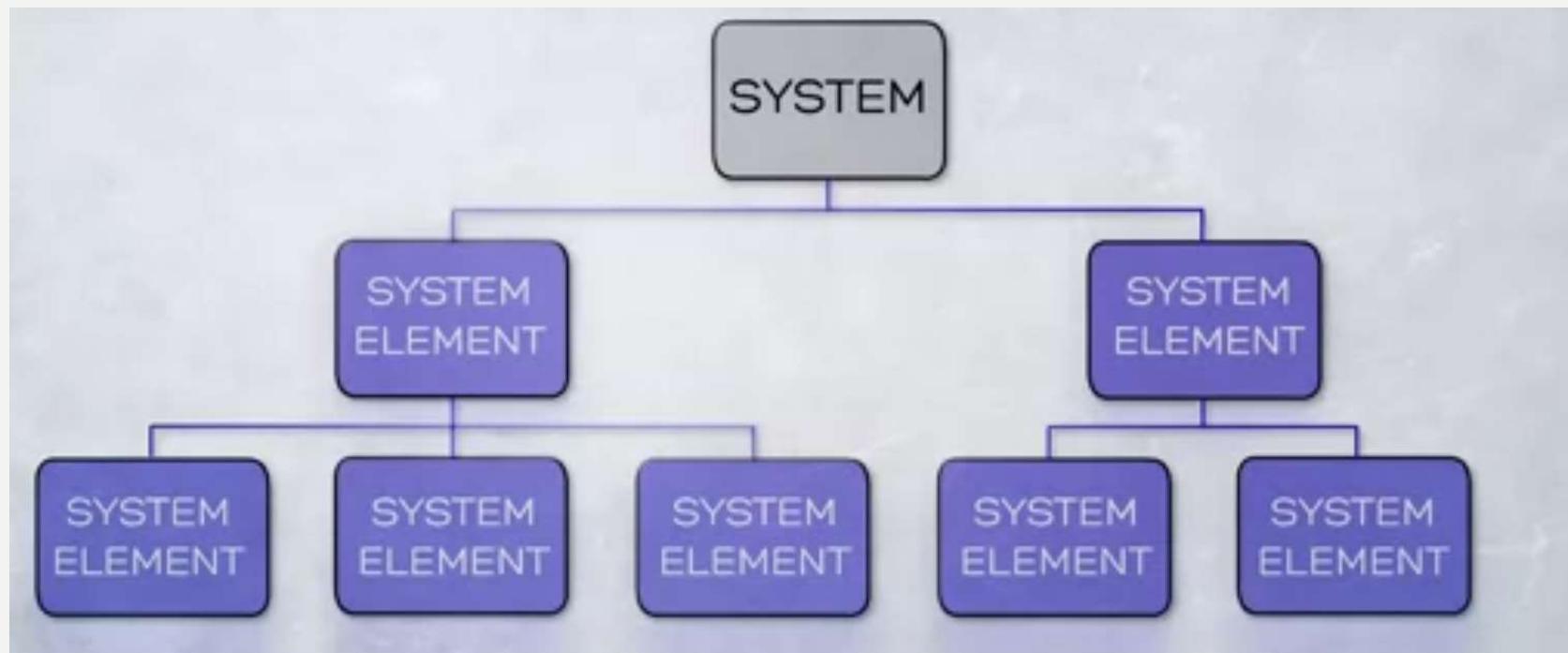
Un système peut être défini selon deux descriptions :

- 1- logique ou fonctionnelle : mission du système : qu'est ce qu'il va faire , comment il va faire , comment il va être tester, sous quelle conditions va-t-il fonctionner quel autre systèmes vont être impliqués avec lui ?
- 2- Physique : les composants du système, comment il va être fabriqué intégré et testé ?



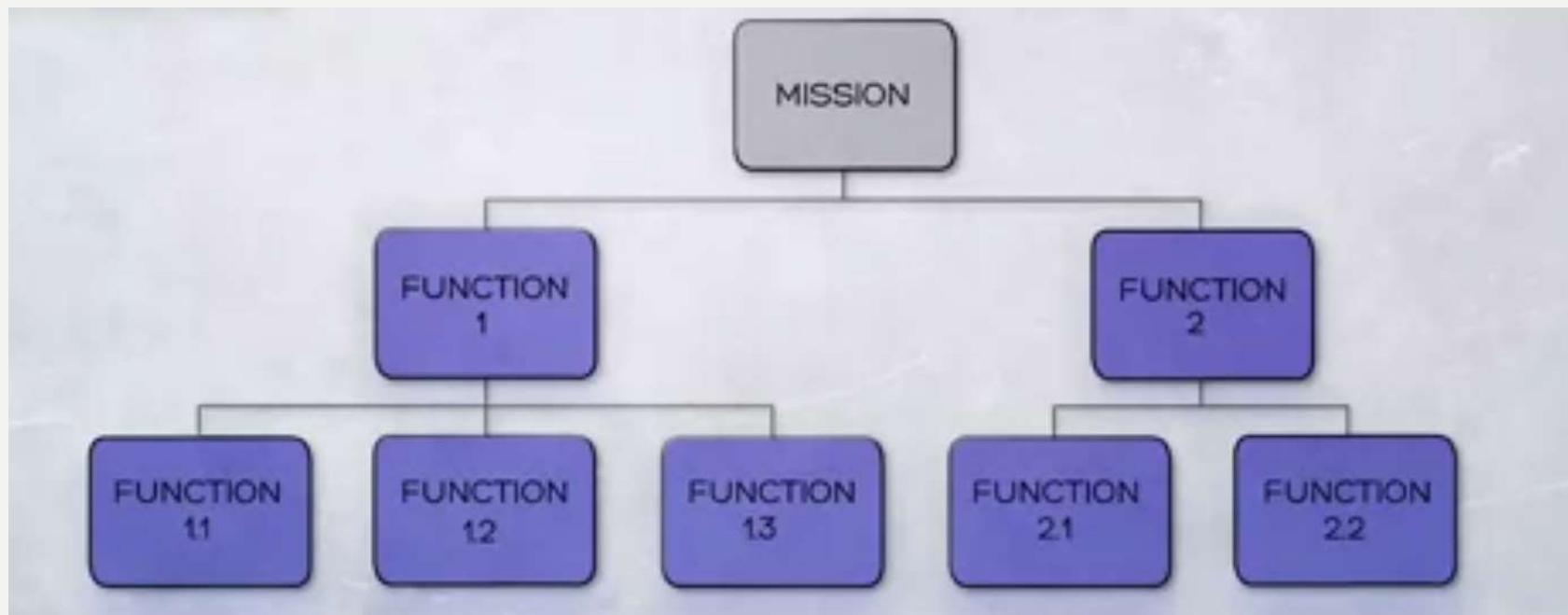
# Description hiérarchique du système

- On peut considérer que le système est une composition hiérarchique d'éléments de système (logique ou physique)

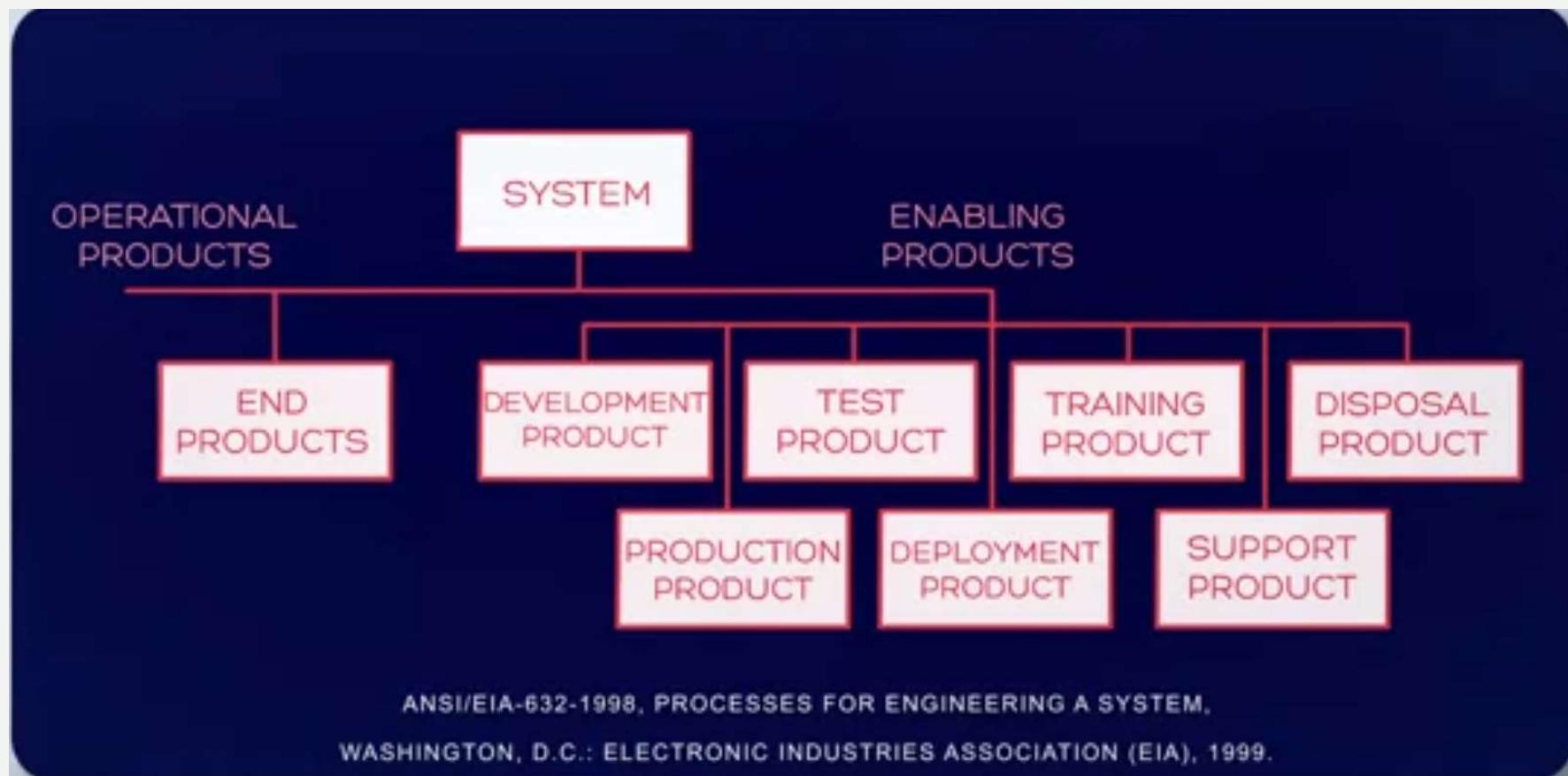


# Architecture Fonctionnelle(logique)

- Dans une description logique d'un système, la mission du système est décomposée en une structure hiérarchique de ses principales fonctions – pour former une hiérarchie fonctionnelle ou une architecture fonctionnelle

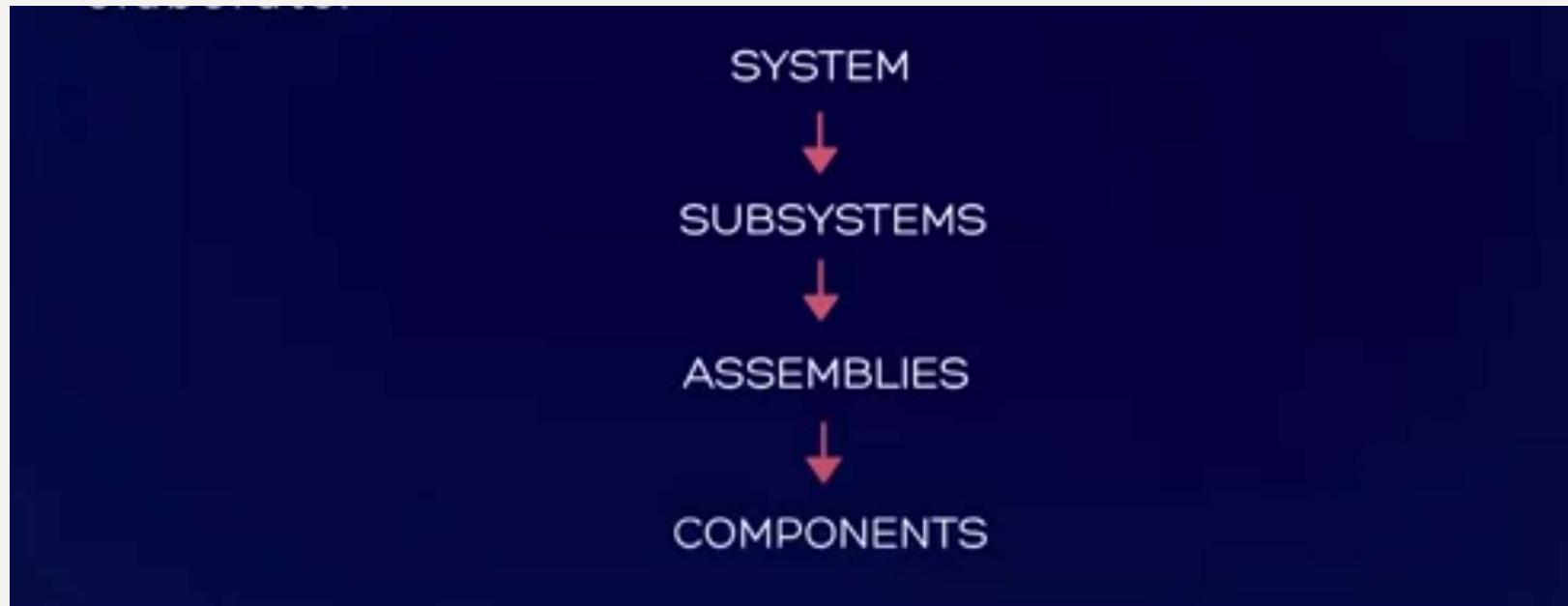


# Architecture Physique (organique)



# Architecture Physique (organique)

- nous utilisons une représentation simple à quatre couches :système, sous-système, composant



# Conclusion



**Remarque : « La description hiérarchique du système est relative »**

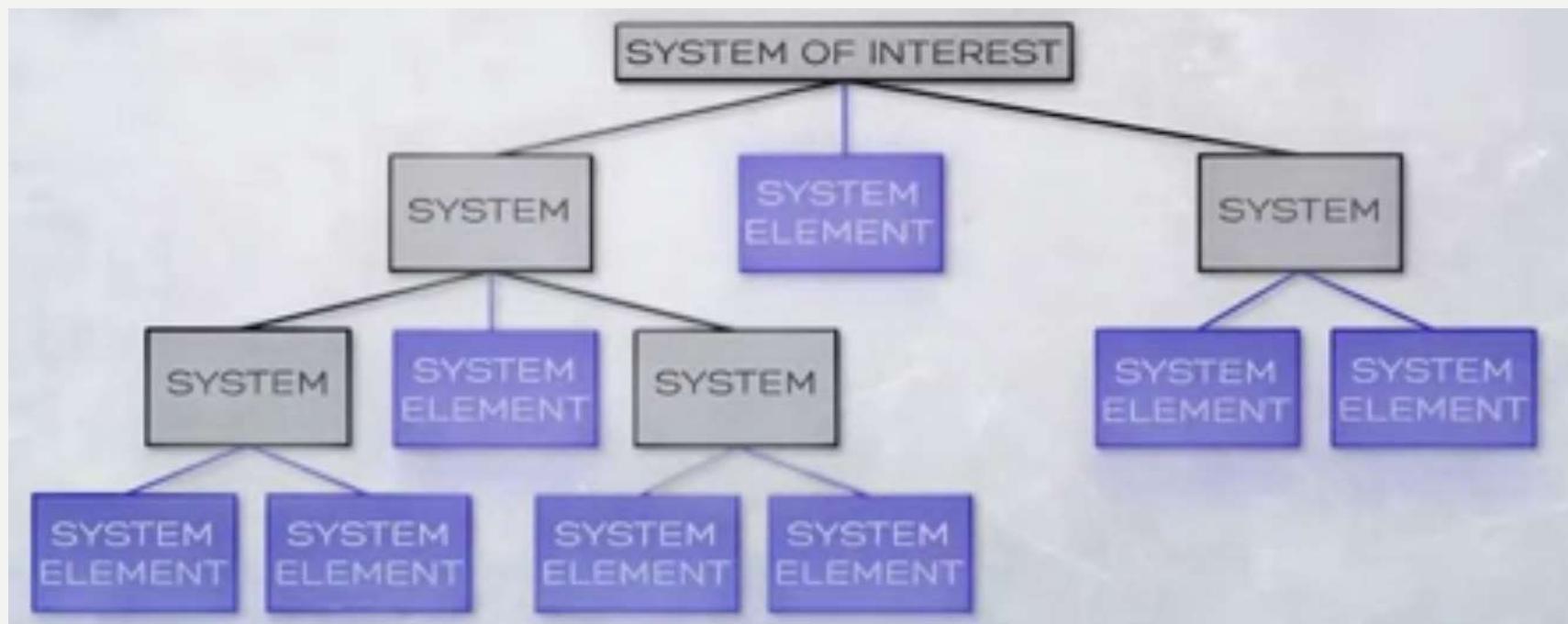
### Exemple

- Un système d'avion contient, entre autres, le **sous-système** moteur, qui peut comprendre des **ensembles** :
- Des réservoirs de carburant,
- Des pompes et des conduites,
- des turbines, des compresseurs,
- des boîtes à vitesses et des pompes hydrauliques
- Le fabricant du moteur peut considérer que le moteur d'avion comme un système , comprenant le carburant, la centrale et le sous-système hydraulique ..etc

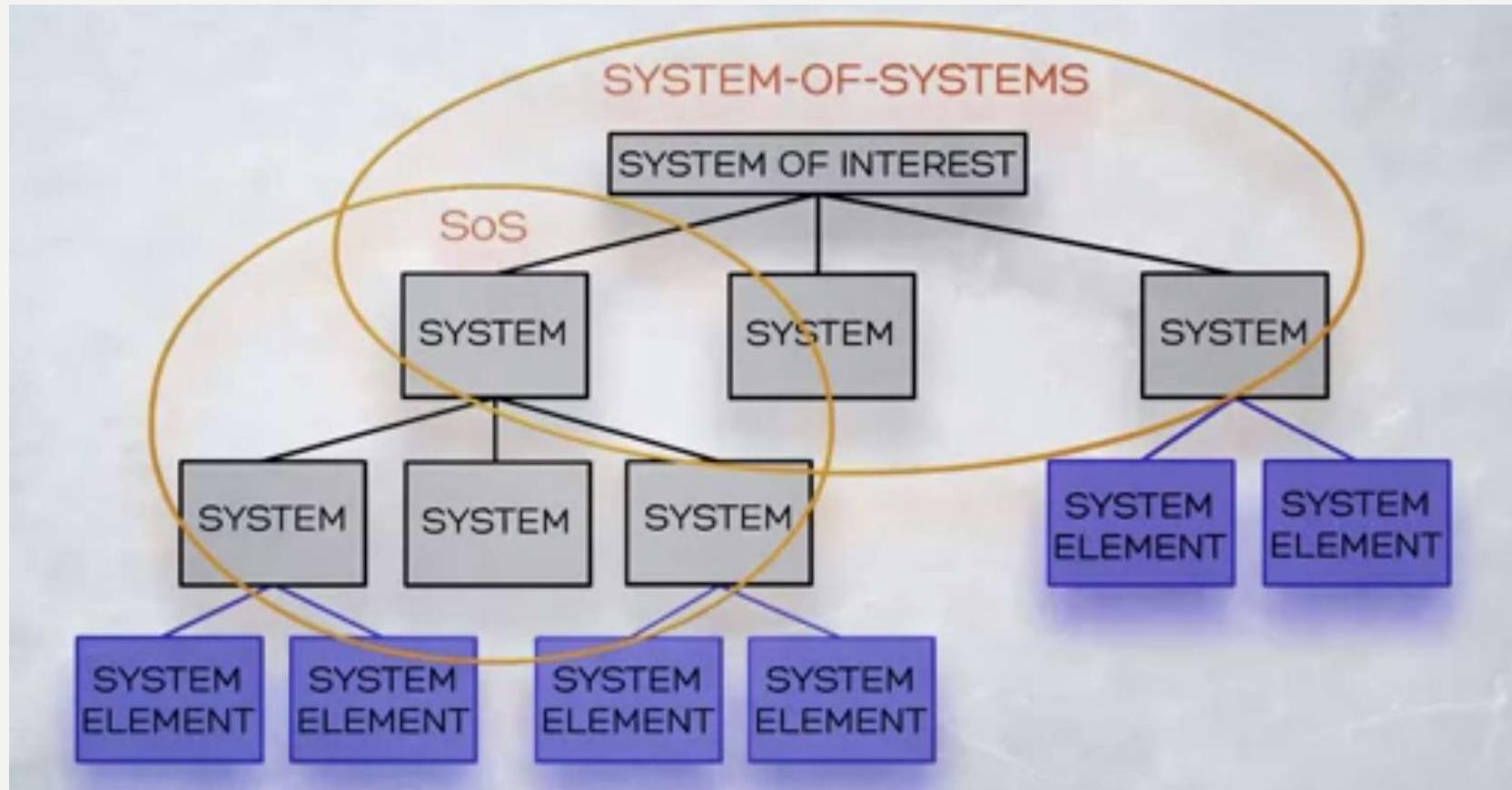
Toutefois, une partie implicite de la définition d'un système, **c'est qu'il doit être capable de se débrouiller seul.** Un moteur n'est donc pas un système mais seulement un élément d'un système (c'est-à-dire un sous-système)

# Hiérarchie du système d'intérêt : soi (system of interest)

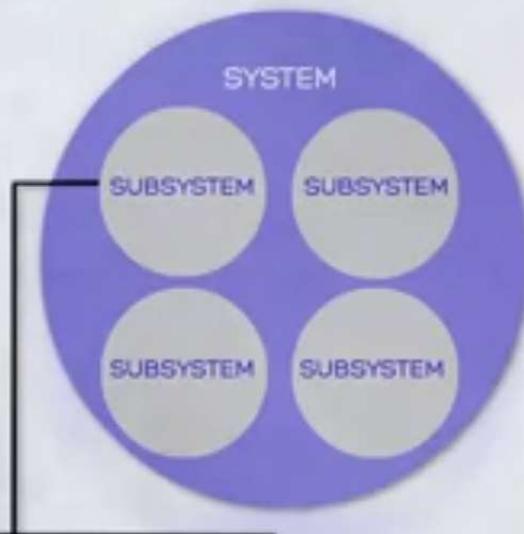
- Il vaut mieux considérer le système SOI pour inclure une combinaison d'éléments de système en interaction, dont certains peuvent être système à part entière



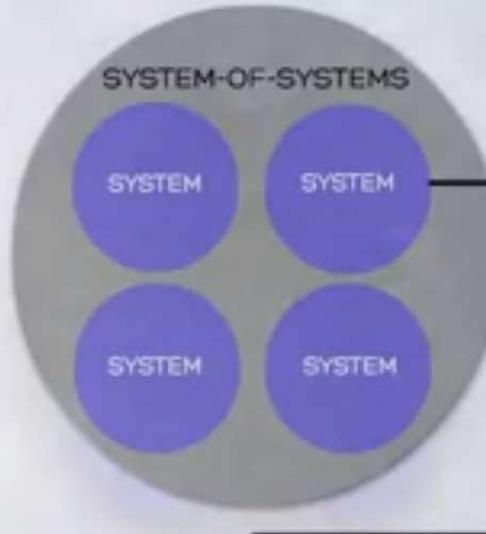
# SYSTÈME DES SYSTÈMES : SOS



## SYSTEM OR SYSTEM-OF-SYSTEMS (SoS)?



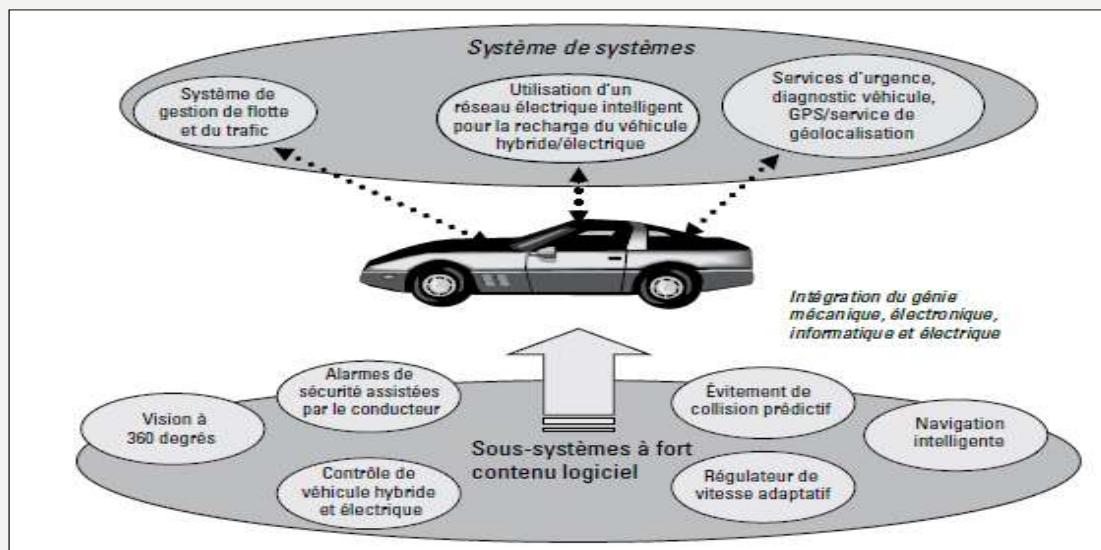
Optimised for the system's purpose  
System is optimised



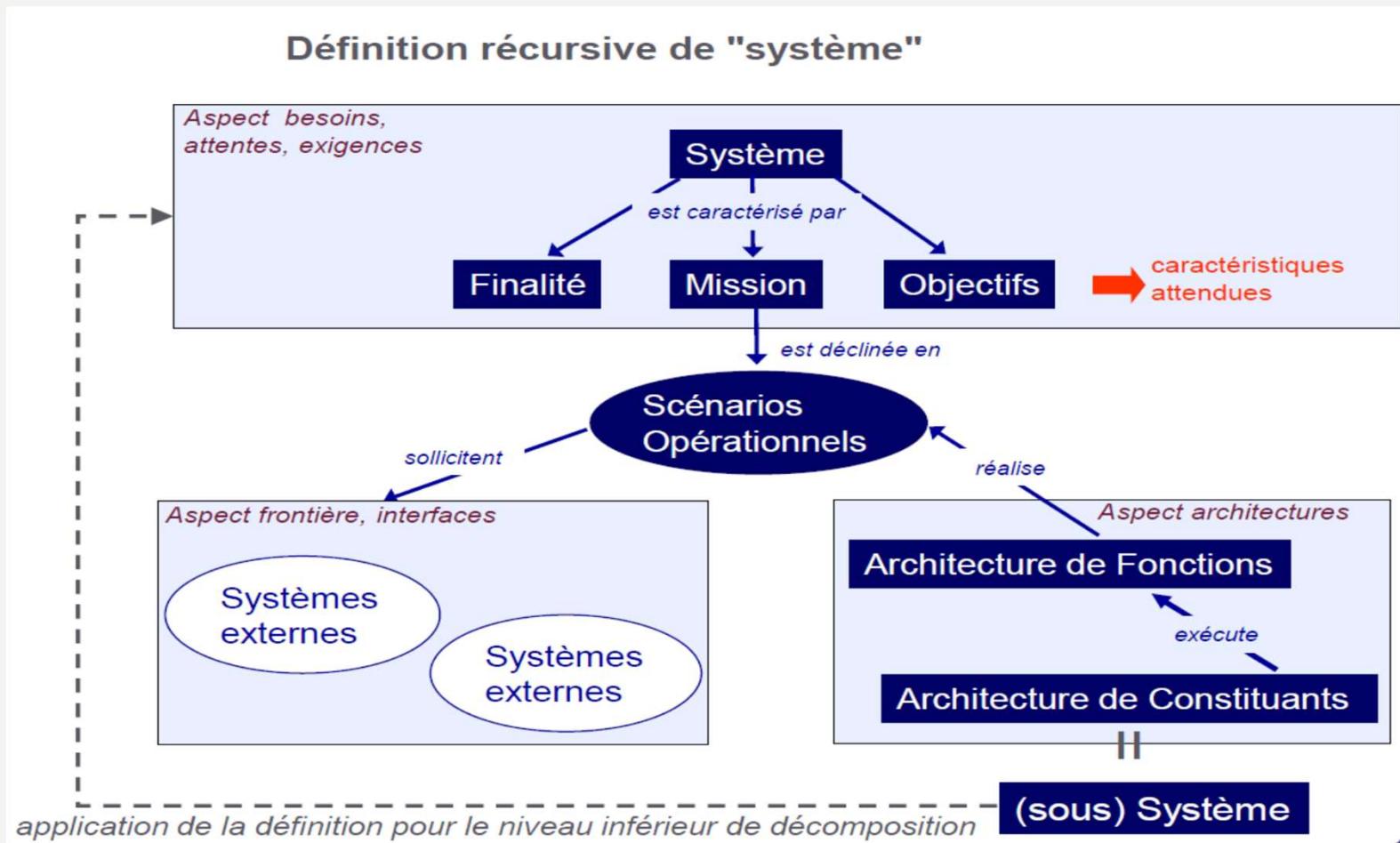
Optimised for their own purposes  
SoS is most likely not optimised

## Exemple

- La conception et la fabrication d'une voiture moderne , qui est véritablement un «*système des systèmes*», : éléments mécaniques, électriques et électroniques, et logiciels.
- Les voitures modernes interagissent aussi avec d'autres systèmes extérieurs à la voiture. : Les services de positionnement, les systèmes de diagnostic véhicule les systèmes de charge des modèles hybrides ou électriques



# Conclusion : Structure d'un Système



# Cycle de vie d'un système/produit

*« Un système a une vie. Il est créé, utilisé, puis éliminé lorsqu'il ne sert plus à rien »*

# Introduction

- La définition d‘un cycle de vie du système permet :
  - ✓ d‘avoir une vision globale et compréhensible du système.
  - ✓ d‘établir un cadre pour répondre aux besoins des parties prenantes PPs d‘une manière ordonnée et efficace.
- Les objectifs et les livrables doivent être définis pour chaque phase du cycle de vie. Les processus et les activités du cycle de vie sont choisis et adaptés pour chaque phase et utilisés pour atteindre les objectifs et les livrables de cette phase‘ (ISO/IEC/IEEE15288 2015).

# Cycle de Vie générique

- Tout au long de la vie d'un système, il y a un certain nombre de phases et d'activités, dont chacune s'appuie sur les résultats de la phase ou de l'activité précédente
- La somme de toutes ces activités est appelée cycle de vie du système
- Un cycle de vie générique du système peut être divisé en quatre grandes phases : phase de pré-acquisition, phase d'acquisition, phase d'utilisation et phase de retrait

Pre-acquisition Phase

Acquisition Phase

Utilisation Phase

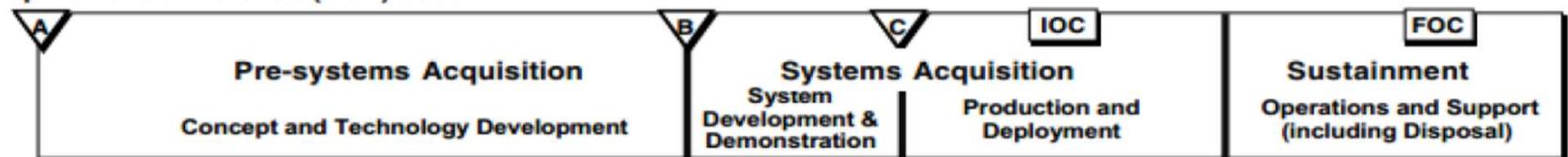
Retirement Phase

# *Les différents points de vue d'un cycle de vie d'un système*

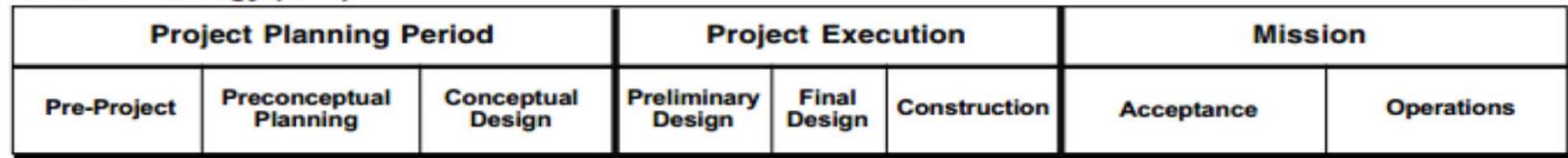
ISO/IEC 15288



US Department of Defense (DoD) 5000.2



US Department of Energy (DoE)



Typical Decision Gates

New Initiative Approval

Concept Approval

Development Approval

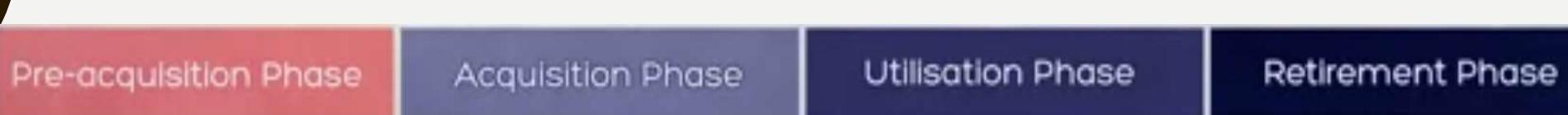
Production Approval

Operational Approval

Deactivation Approval

# La phase de pré-acquisition

- Le cycle de vie commence par la phase de pré-acquisition avec une idée pour un système générée par suite de la planification des activités
- Les besoins opérationnels sont confirmés et appuyés par une analyse de rentabilité
- Cette phase veille à ce que seuls les projets réalisables et rentables soient transférés à la phase de l'acquisition



# La phase d'acquisition

- La phase d'acquisition vise à mettre le système au service de l'organisation
- Le système est défini en termes de:
  - exigences opérationnelles
  - exigences des PPS
  - exigences du système
- Un entrepreneur est alors normalement engagé pour élaborer/livrer le système



# La phase d'utilisation

- La phase d'exploitation ou la phase opérationnelle est la phase pendant laquelle le système est exploité dans les environnements prévus en fournissant les services prévus avec une efficacité opérationnelle continue. Les modifications du système sont souvent prévues le long du cycle de vie du système. Ces modifications améliorent les capacités du système

Pre-acquisition Phase

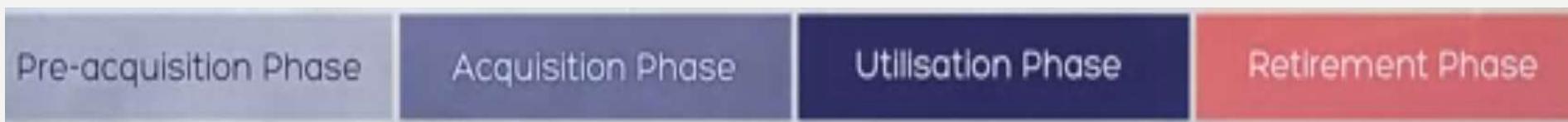
Acquisition Phase

Utilisation Phase

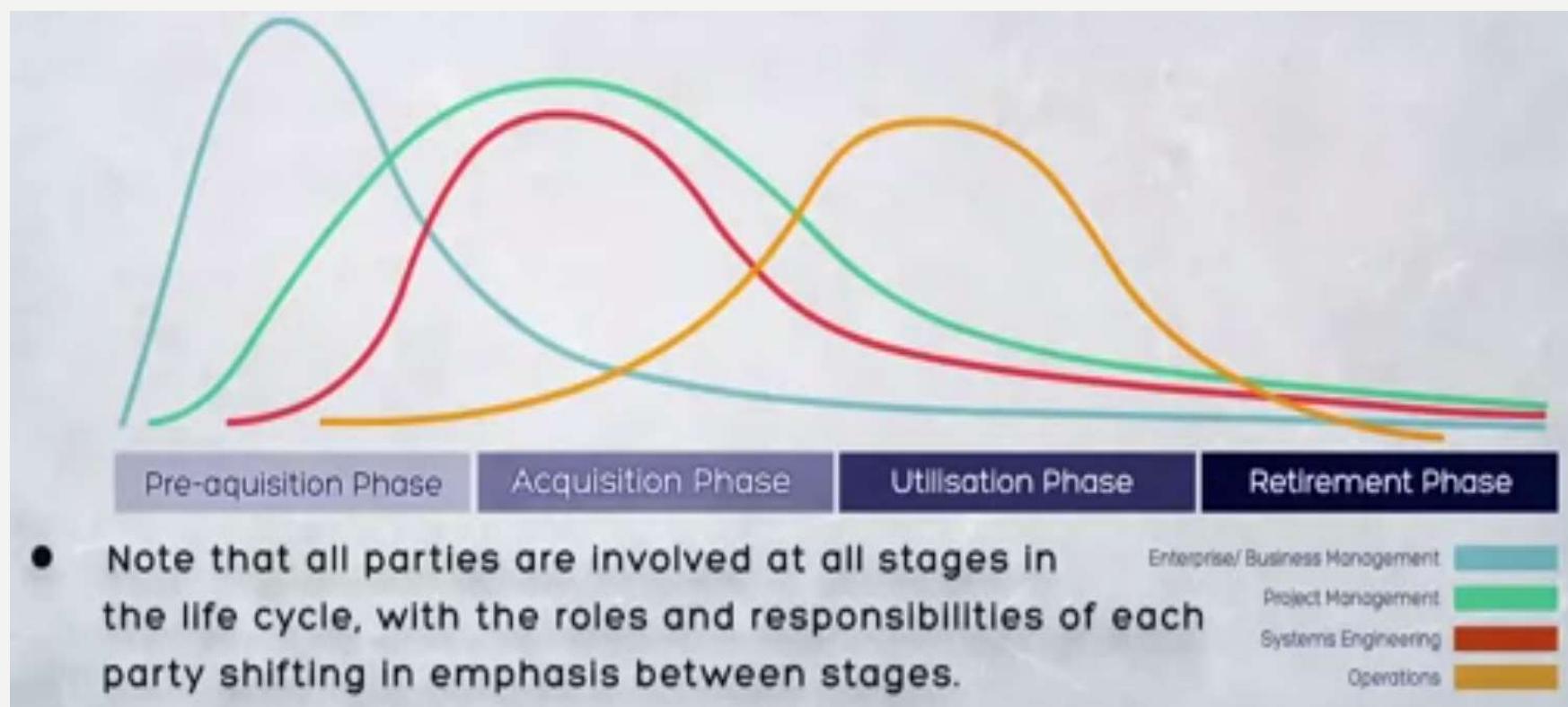
Retirement Phase

# La phase de retrait

- Le système est en service pendant la phase d'utilisation jusqu'à ce que :
  - L'entreprise n'a plus besoin du système, ou
  - Il ne peut plus remplir les fonctions que l'organisation exige de lui, ou,
  - Il n'est plus rentable de le maintenir en service.
- Si le besoin opérationnel pour la capacité existe toujours dans l'organisation. La fin d'un cycle de vie du système marque le début d'un autre et le processus recommence

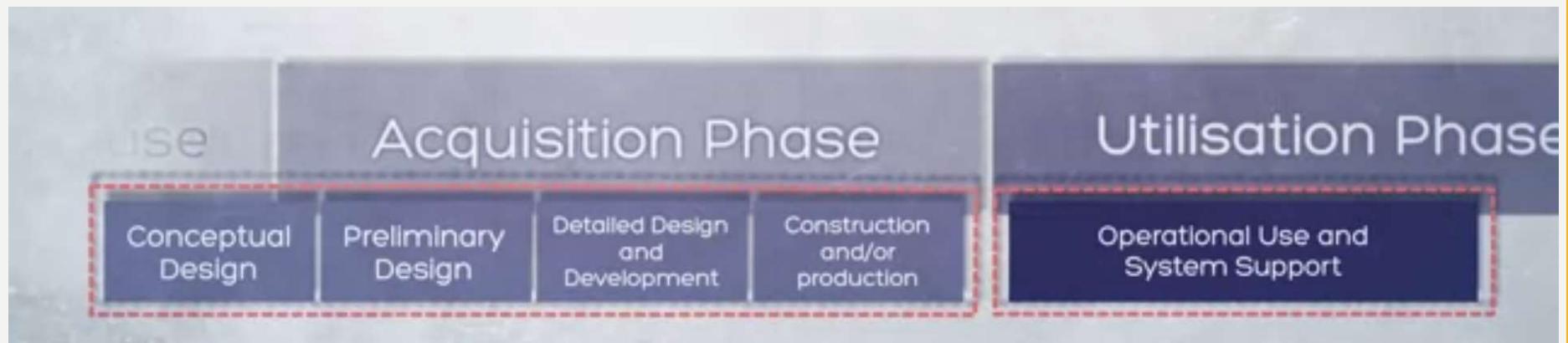


# La responsabilité des parties opérantes le long du cycle de vie du système

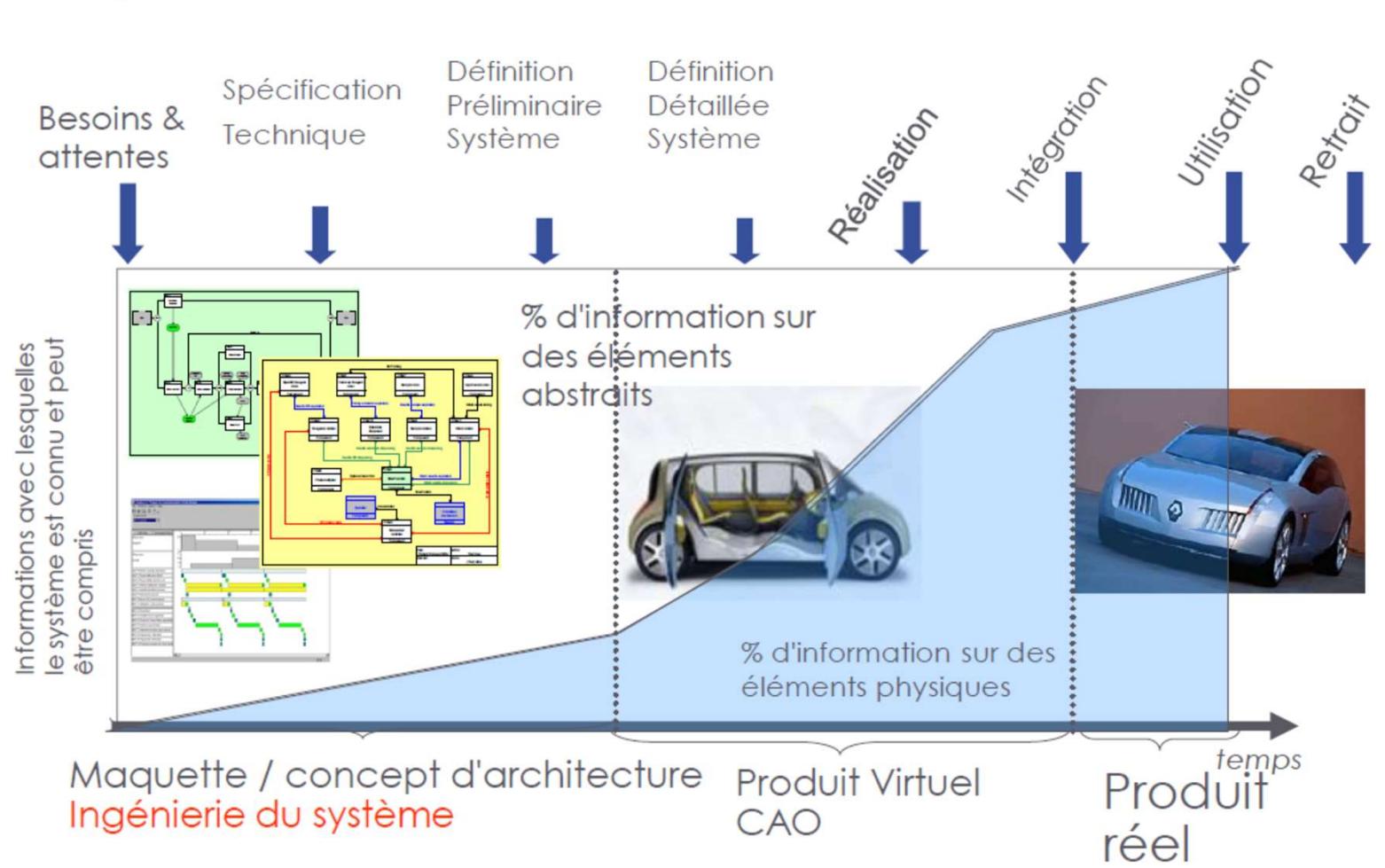


# Ingénierie de système et cycle de vie

- L'ingénierie des systèmes est principalement liée à la phase d'acquisition du cycle de vie du système et, dans une moindre mesure, à la phase d'utilisation
- Ces deux phases comprennent un certain nombre d'activités



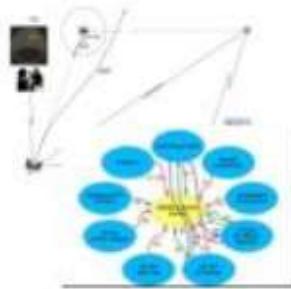
# Exemple



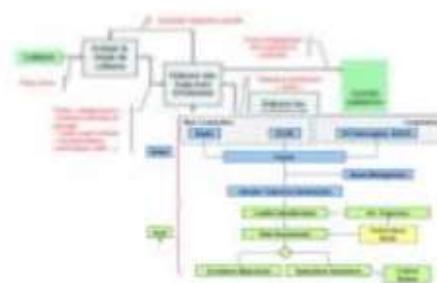
# Midcas, un programme pour faire voler les drones dans le trafic aérien

→ Couvre tous le cycle de vie d'un produit (exemple MIDCAS)

Définition des besoins



Spécification technique



Définition préliminaire



Définition détaillée



Réalisation



Intégration

Utilisation

Retrait

Concept / architecture

Maquette/conception

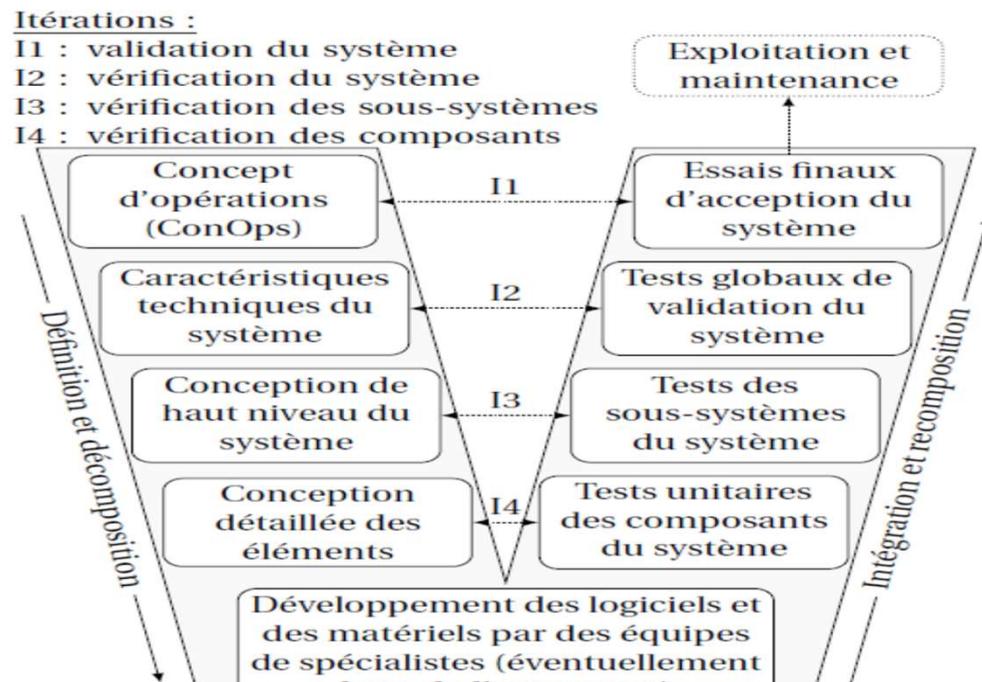
produit

# Approches d'analyse des systèmes complexes

*Modèle en « cascade », en « V » et en « spirale »*

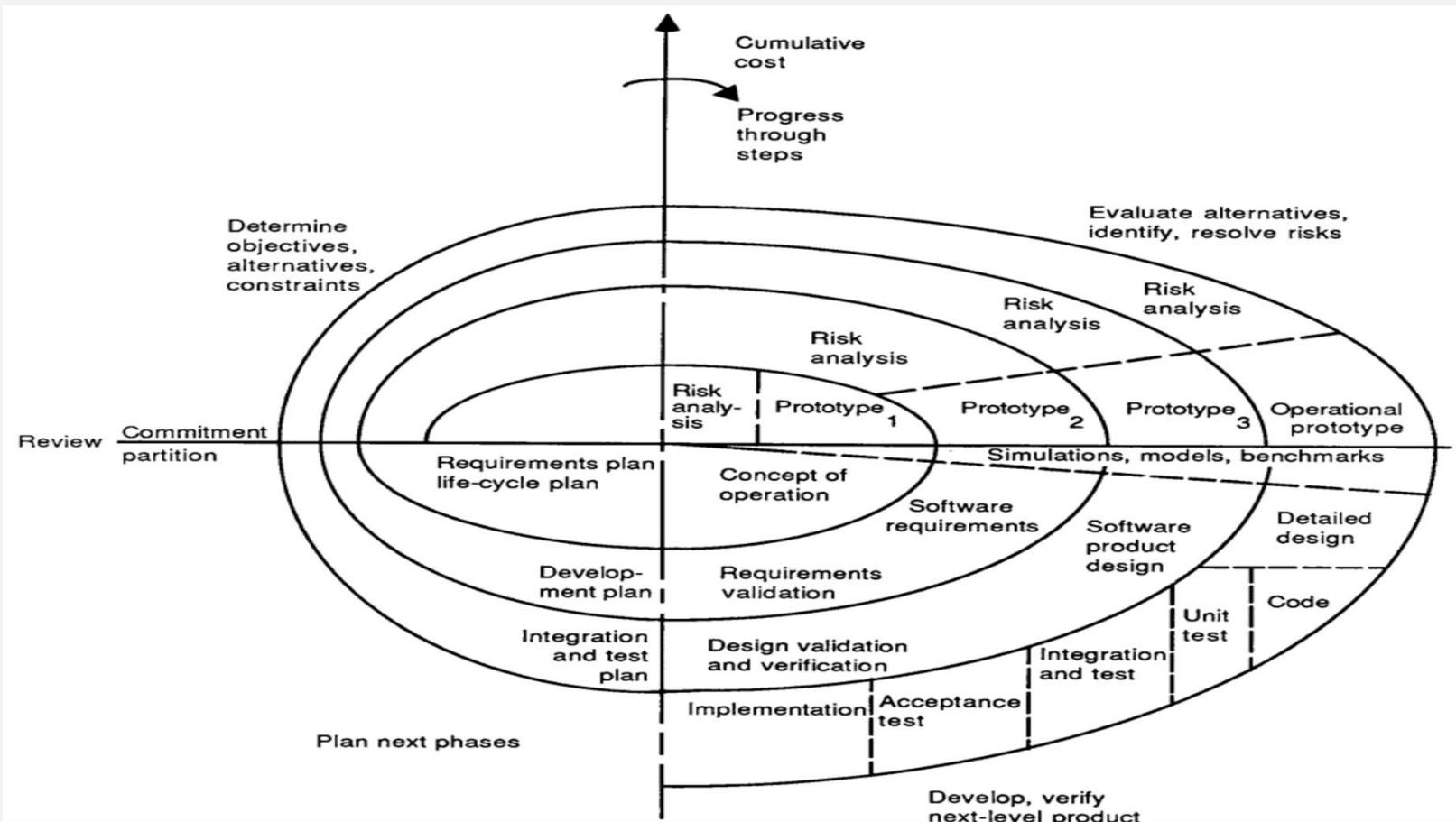


(a) En cascade

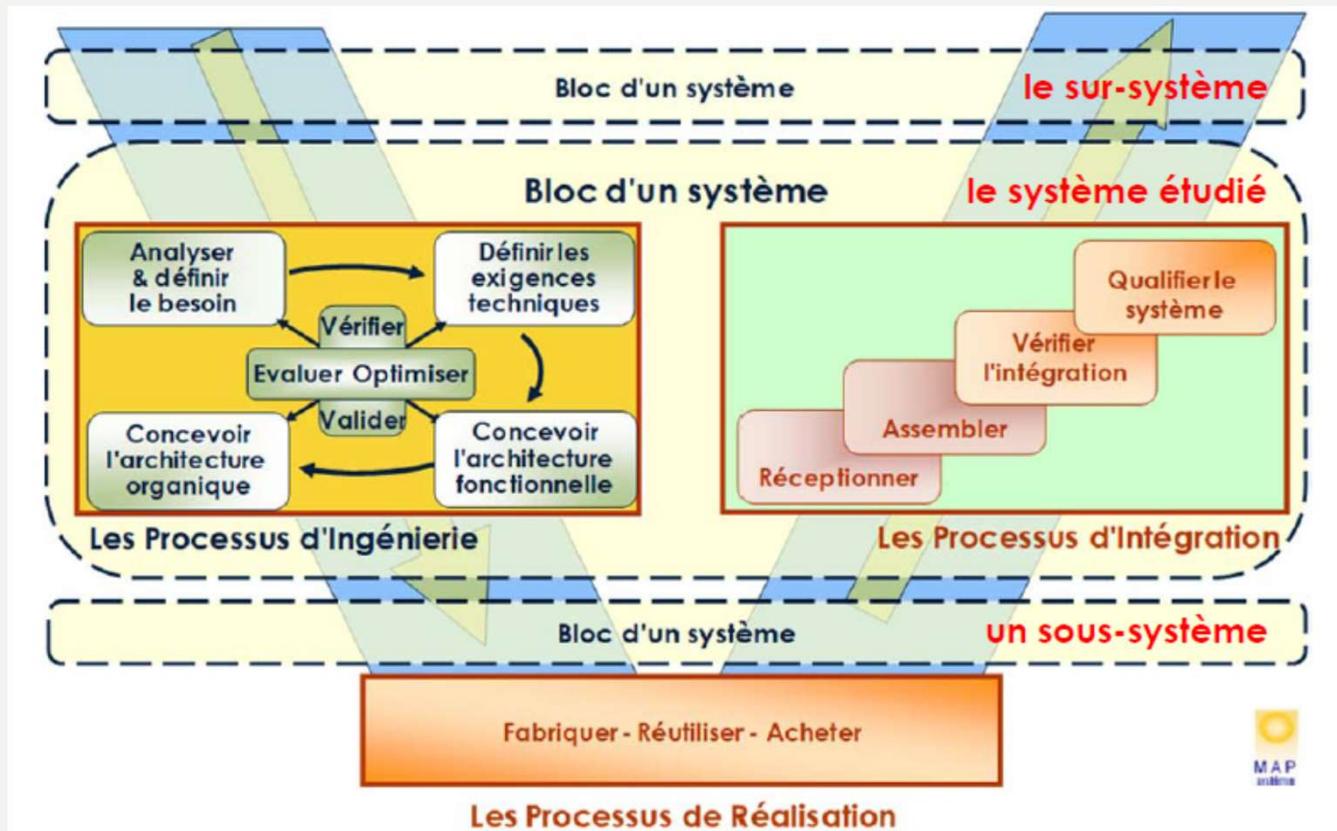


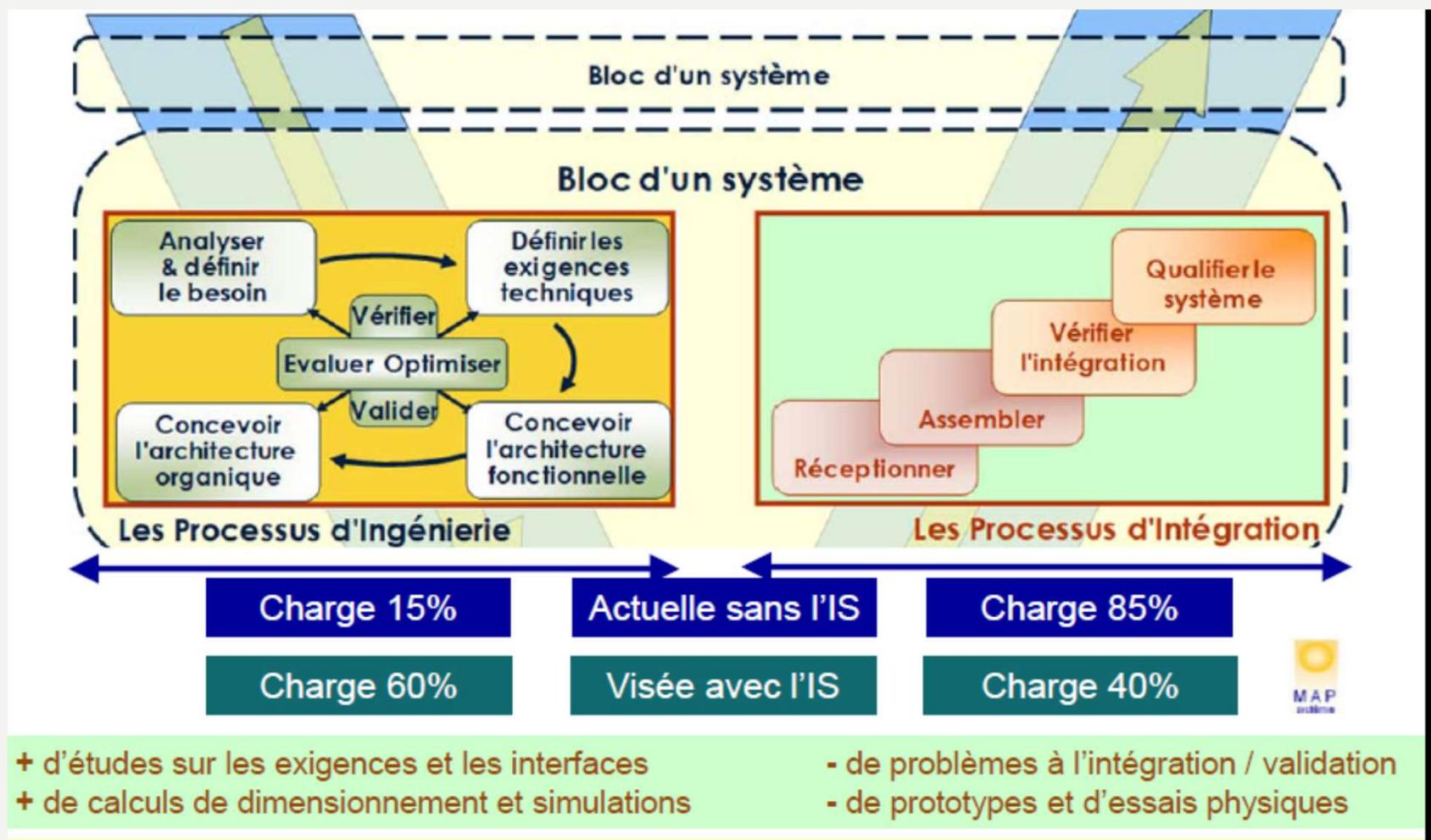
(b) En « V » (Source : IBM)

## « Modèle en spirale »



# Ingénierie système vs Cycle en V





# **CHAPITRE 2 : PROCESSUS, METHODE ET OUTIL DE L'IS**

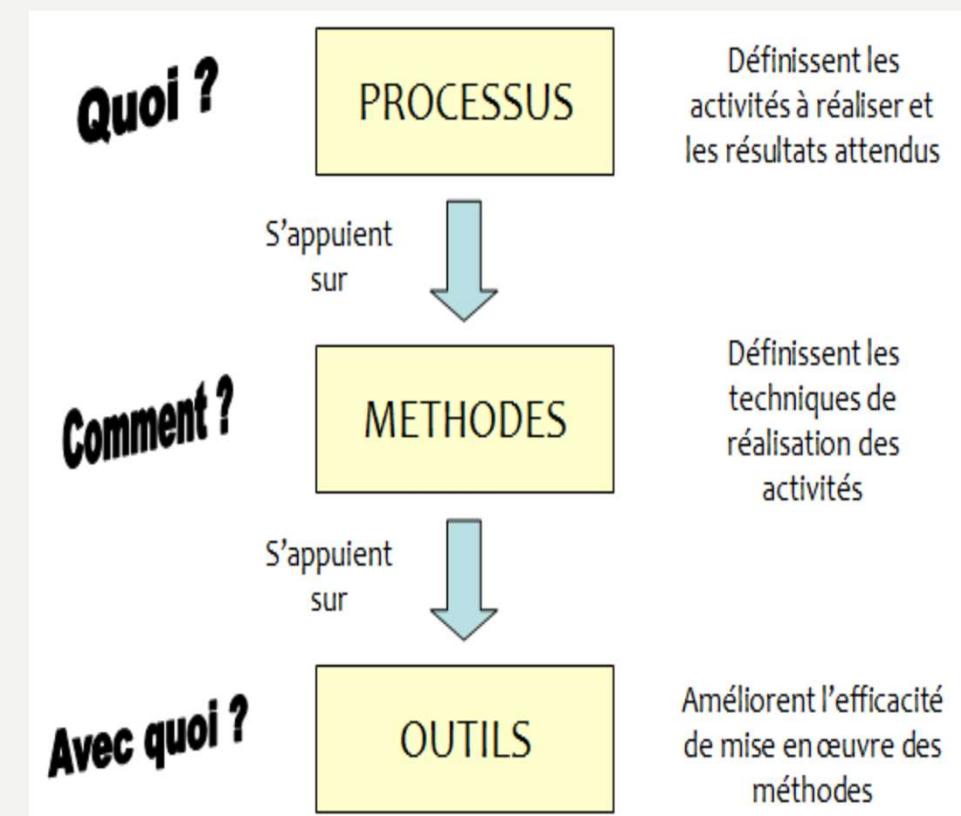
# **PLAN**

- Normes de l'IS,
- Méthode d'intervention

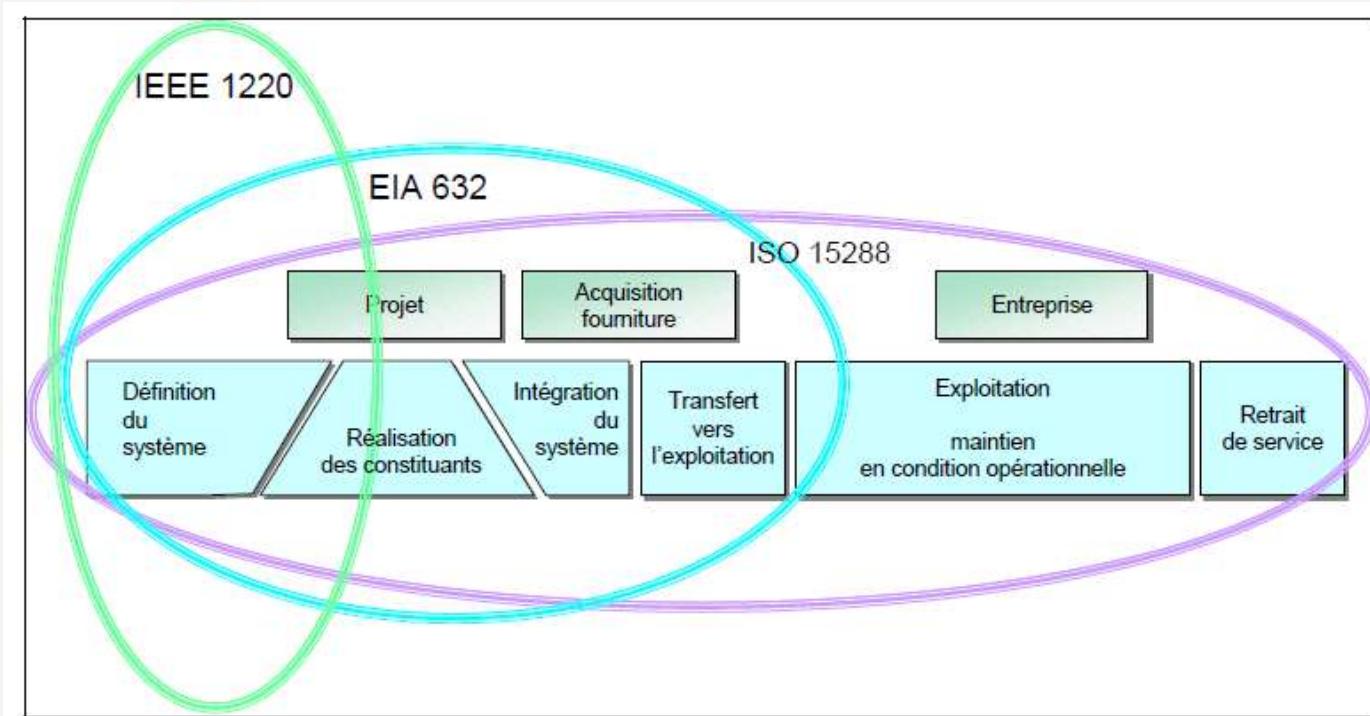
# Normes de l'IS

# INTRODUCTION

- **Les normes d'IS** décrivent les pratiques du métier en termes de processus et d'activités de manière invariante par rapport aux domaines d'application de l'ingénierie système.
- **Les méthodes d'ingénierie système** fournissent des démarches techniques pour réaliser ces activités générales.
- La mise en œuvre des processus et des méthodes est assistée par des **outils**, aujourd'hui très généralement informatisés.

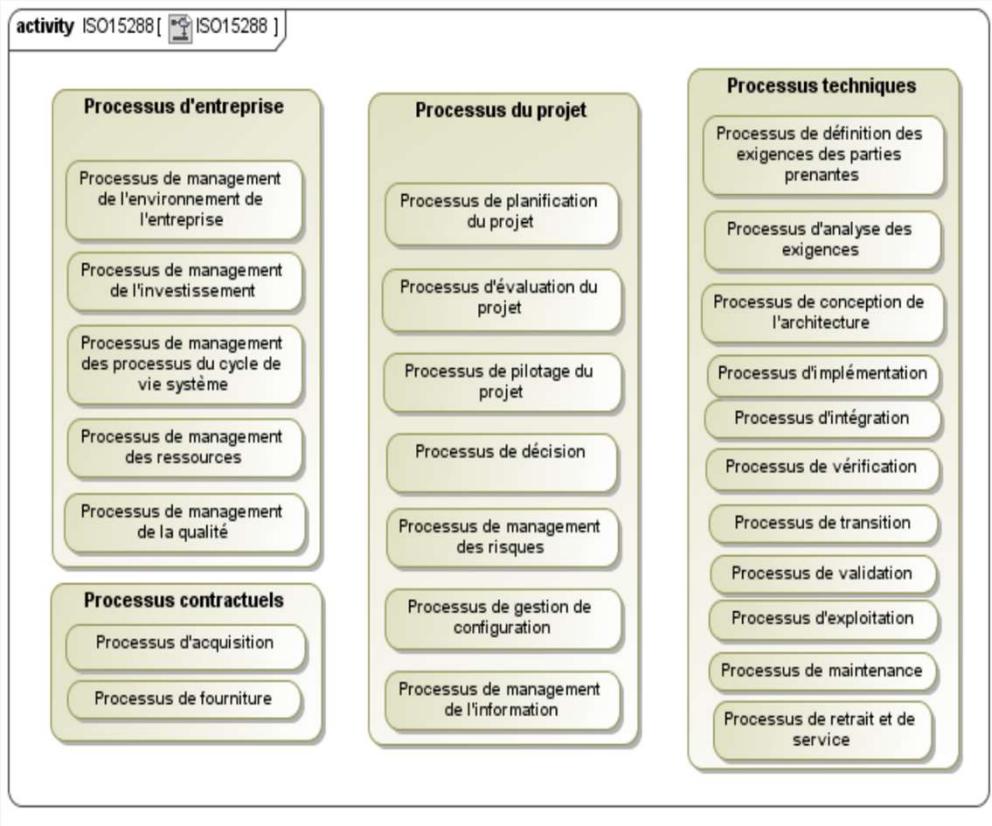


# LES NORMES DE L'INGÉNIERIE SYSTÈME



L'IS retient comme invariant les processus normalisés du cycle de vie des systèmes  
NORMES: ISO 15288, EIA 632, IEE1220 (AFIS)

# ISO 15 288



- **ISO 15 288 : Systems engineering – System life cycle processes, 2003**
- cette norme de l'ISO étend les processus techniques à tout le cycle de vie du système (elle couvre ainsi les processus d'exploitation, de maintien en condition opérationnelle et de retrait de service).
- La norme s'applique à l'ingénierie des systèmes contributeurs qui ont leur propre cycle de vie (systèmes de fabrication, de déploiement, de soutien logistique, de retrait de service) :

# ISO 15 288

En ingénierie système, tout projet implique la réalisation de divers types d'**activités** :

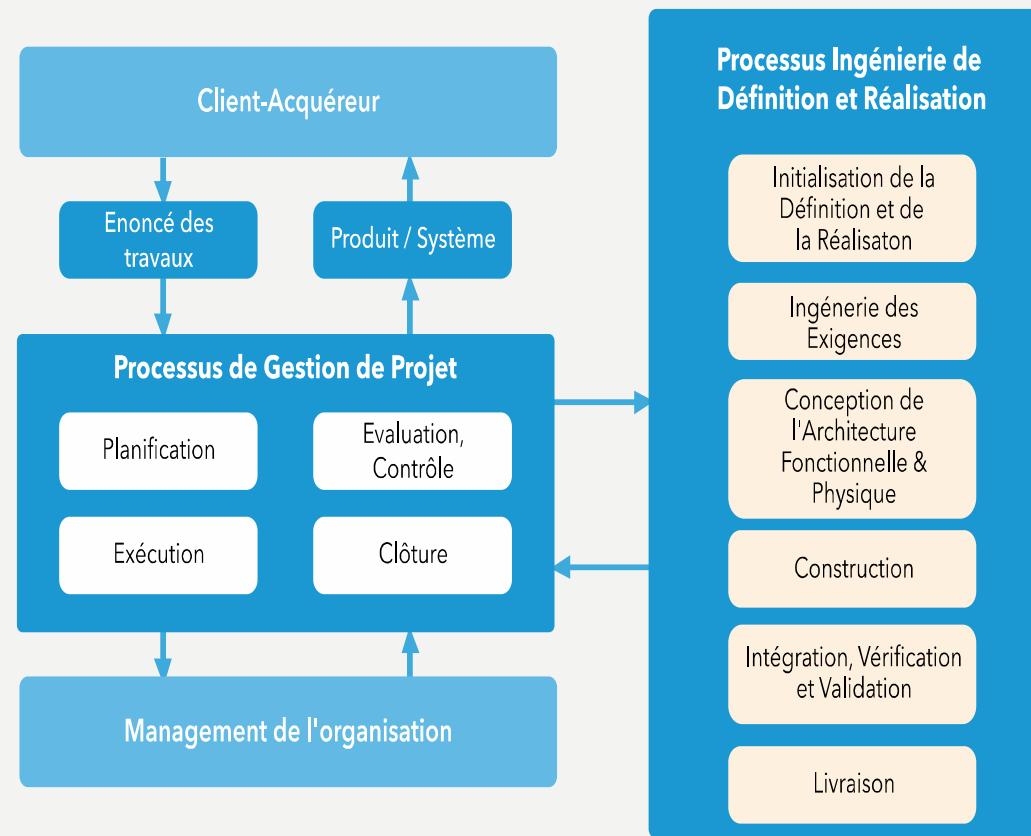
- ❑ activités techniques d'analyse du problème, de conception de la solution, d'intégration, de vérification et de validation,
- ❑ activités de management de ces activités techniques,
- ❑ activités de gestion des relations contractuelles entre clients et fournisseurs (maître d'ouvrage, maître d'œuvre, sous-traitants).

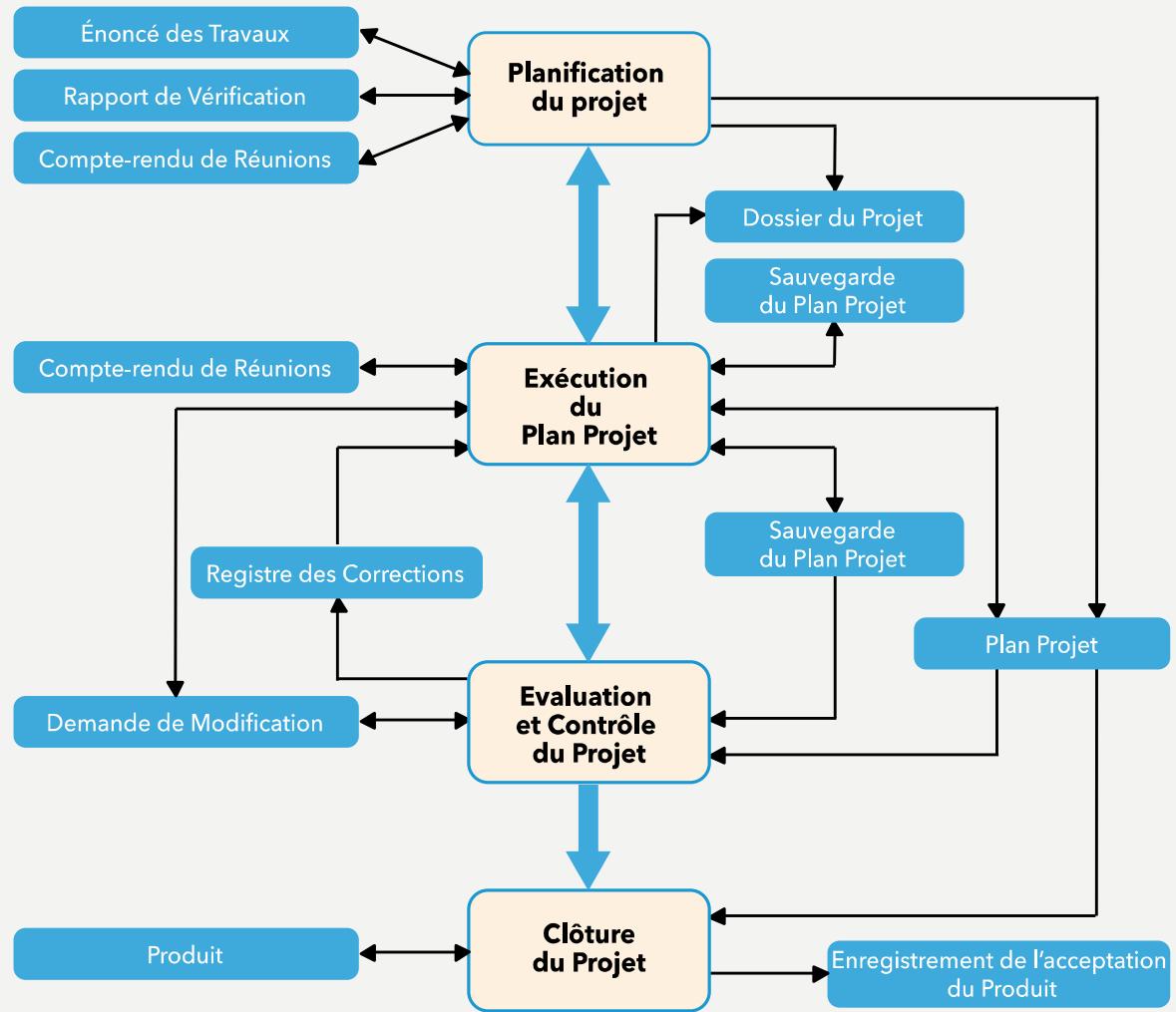
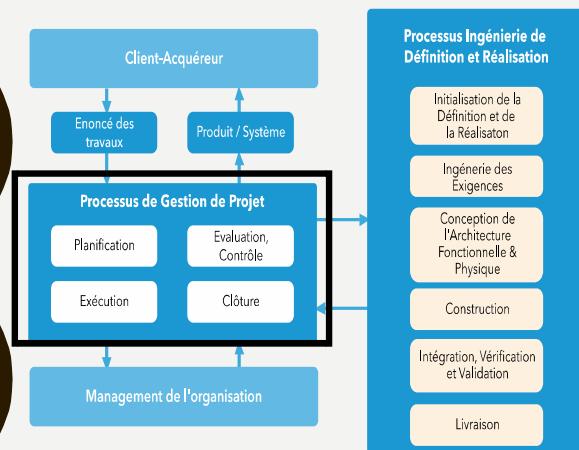
Ces trois types d'activités sont respectivement regroupés en **processus techniques, processus de management et processus contractuels**. Les activités de support communes aux différents projets réalisés par une entreprise, forment une quatrième classe de processus : les **processus d'entreprise**.

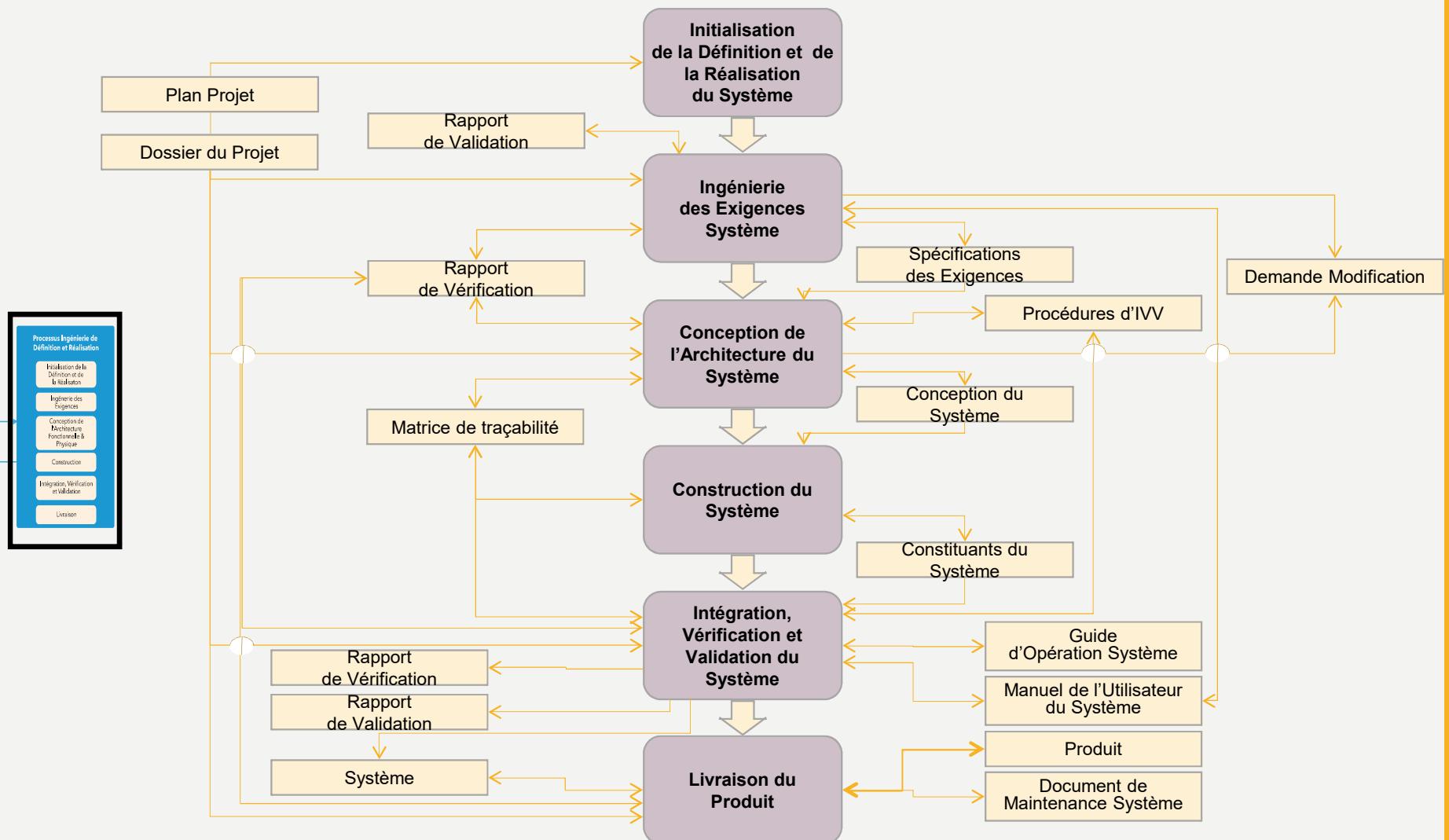
Du fait de l'invariance des types d'activité à réaliser, ces processus sont considérés aujourd'hui comme des fondements génériques pour toute mise en œuvre de l'ingénierie système.

# ISO 29110 : IS et Système Projet :

ingénierie des systèmes et du logiciel — profils de cycle de vie pour très petits organismes (TPO)







# Les processus techniques retenus pour l'IS

## Processus de définition des exigences des parties prenantes (Besoin des PPS)

1 - Identifier les parties prenantes qui ont un intérêt vis-à-vis du système pendant son cycle de vie

2 - Susciter et découvrir les exigences des parties prenantes (en terme de besoin, de manque, de désir, d'attente et de contraintes etc.)

Au total la norme fixe 9 activités

## Processus d'analyse des exigences

I Définir les frontières fonctionnelles du système, exprimées sous la forme du comportement attendu et de ses propriétés (vis-à-vis des utilisateurs, du comportement de l'environnement, des contraintes d'interface etc.

2 – Définir chaque fonction que le système doit exécuter, les critères qualitatifs exigés du système et des opérateurs etc.

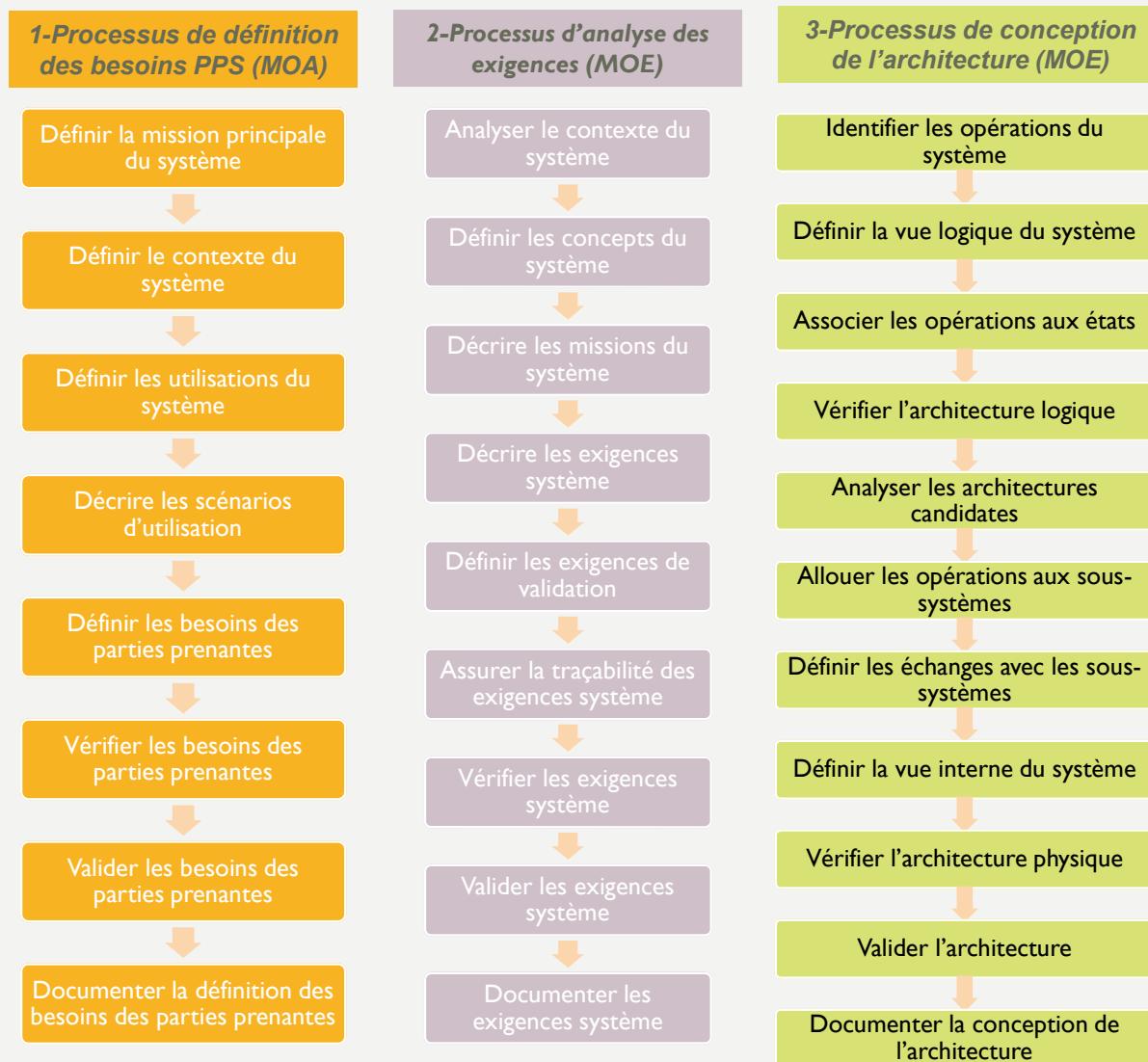
Au total la norme fixe 8 activités

## Processus de conception de l'architecture

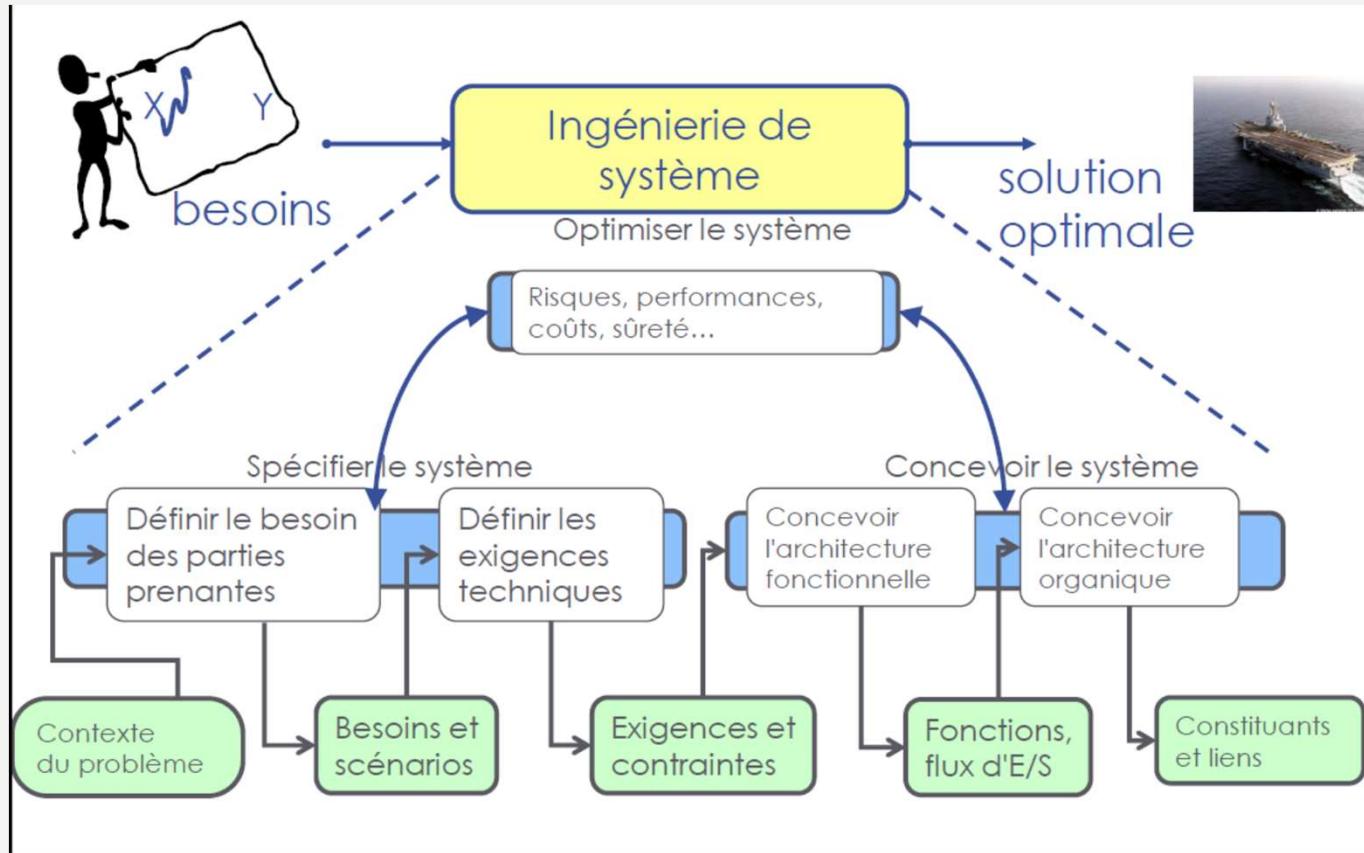
1 – définir des architectures logiques appropriées (relatives aux fonctions et performances, aux services et exigences temporelles, aux flux de données etc.

2 – Grouper ou séparer les fonctions du système identifiées lors de l'analyse des exigences des PPS, et les allouer aux éléments du système etc.)

Au total la norme fixe 10 activités



# Les processus d'ingénierie



# Méthode D'intervention

# ARCHITECTURE DU SYSTÈME EN 2 VUES

## Vue fonctionnelle

- Externe : Le système fournit à son environnement des services ( par l'exécution de fonctions de service )
- Interne : Le système est vu comme un arrangement dynamique de fonctions ("modules logiques") + leurs interfaces et les flux de données et de contrôle qui conditionnent l'exécution.

### Architecture fonctionnelle (logique)

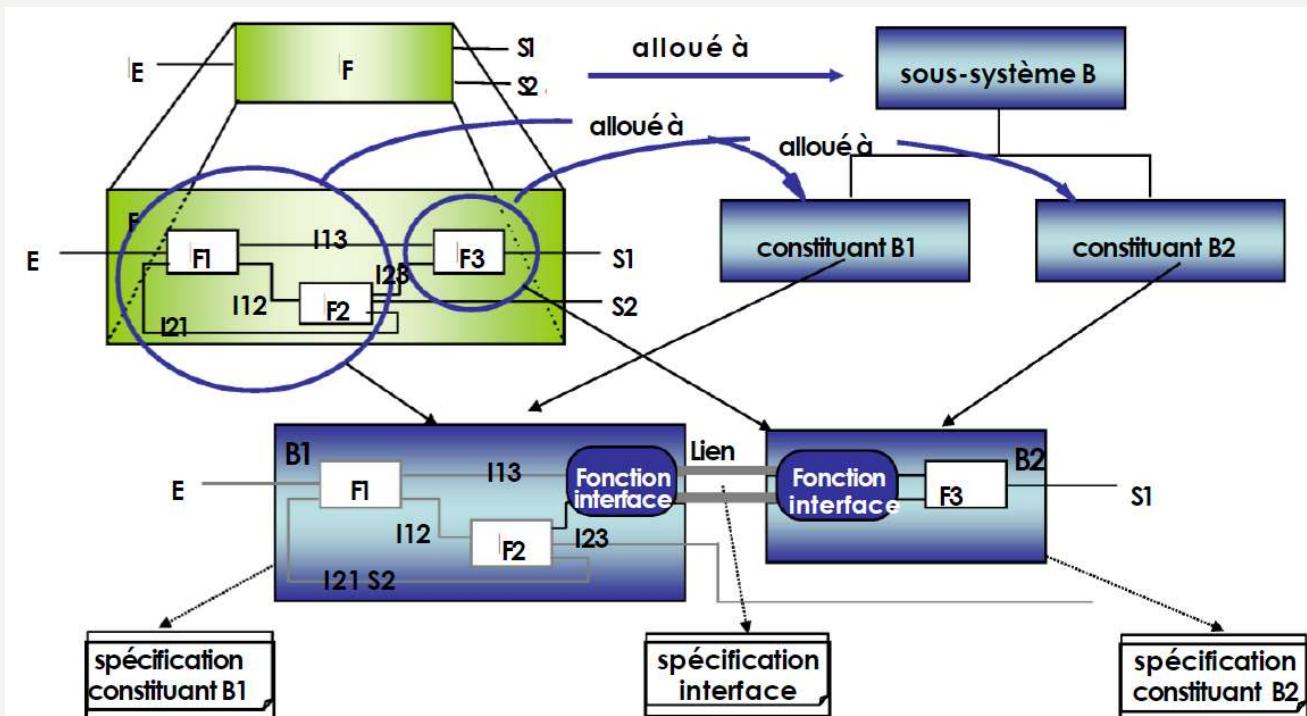
## Vue organique

- Le système est un ensemble d'organes (ou composants) agencés avec des connections physiques, répondant aux exigences de l'architecture fonctionnelle et aux exigences non fonctionnelles

### Architecture organique

# PROJECTION D'UNE VUE SUR UNE AUTRE

- l'arrangement des fonctions
- l'allocation des fonctions aux composants (constituants)
- la spécification des composants et de leurs interfaces

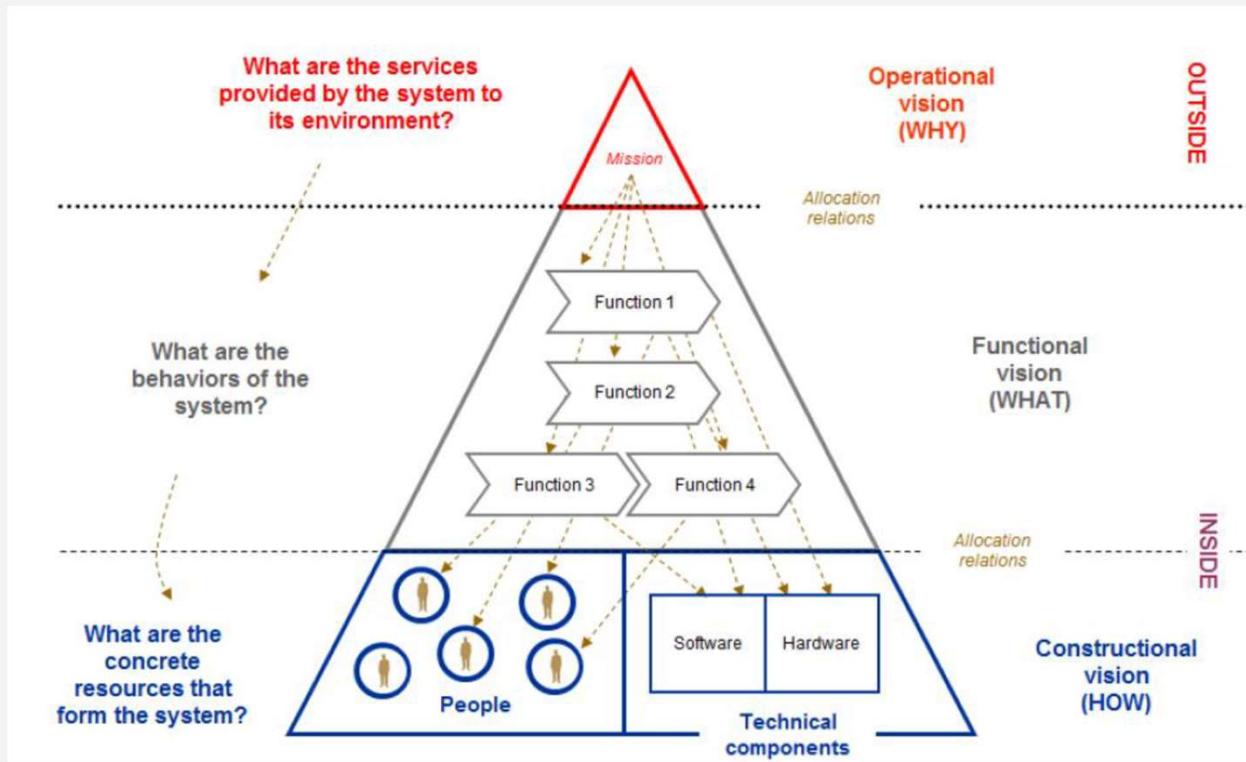


# CESAM SYSTEMS ARCHITECTING METHOD

- CESAM Systems Architecting Method est un *framework* d'architecture et de modélisation développé par CESAMES™;
- Elle est conforme aux standards de l'INCOSE ;
- Les principaux éléments de l'architecture sont :
  - ✓ Une organisation hiérarchique des préoccupations (Quoi ? / Qui ?/ Ou ?), représentée par une "Pyramide d'Architecture Système"
  - ✓ Une organisation en 9 vues du système, représentée par une matrice
  - ✓ Une démarche guidant l'architecture du système, représentée par un processus

# ARCHITECTURE PYRAMID (CESAM)

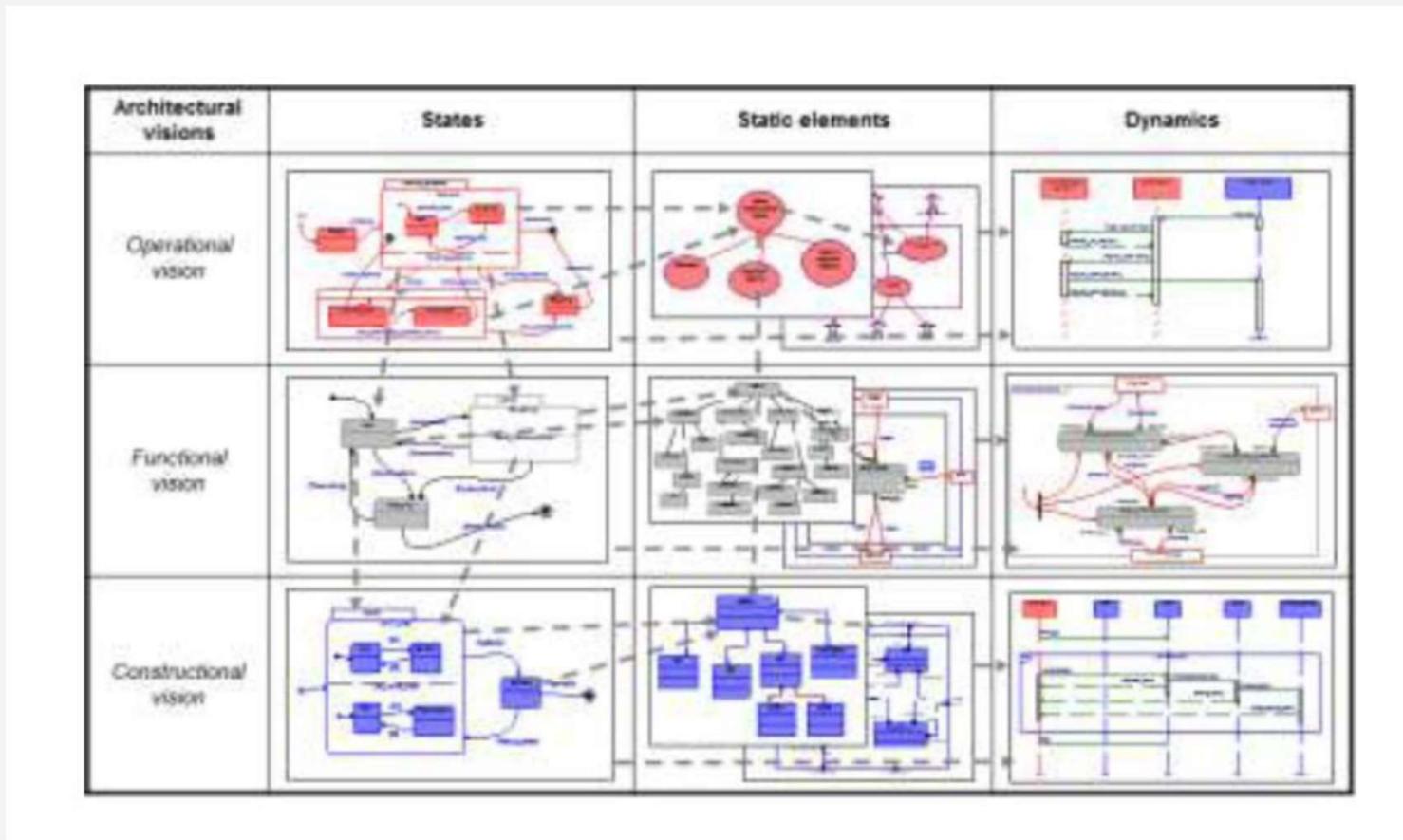
- Le principal intérêt de cette approche est qu'elle est conforme aux standards de l'INCOSE



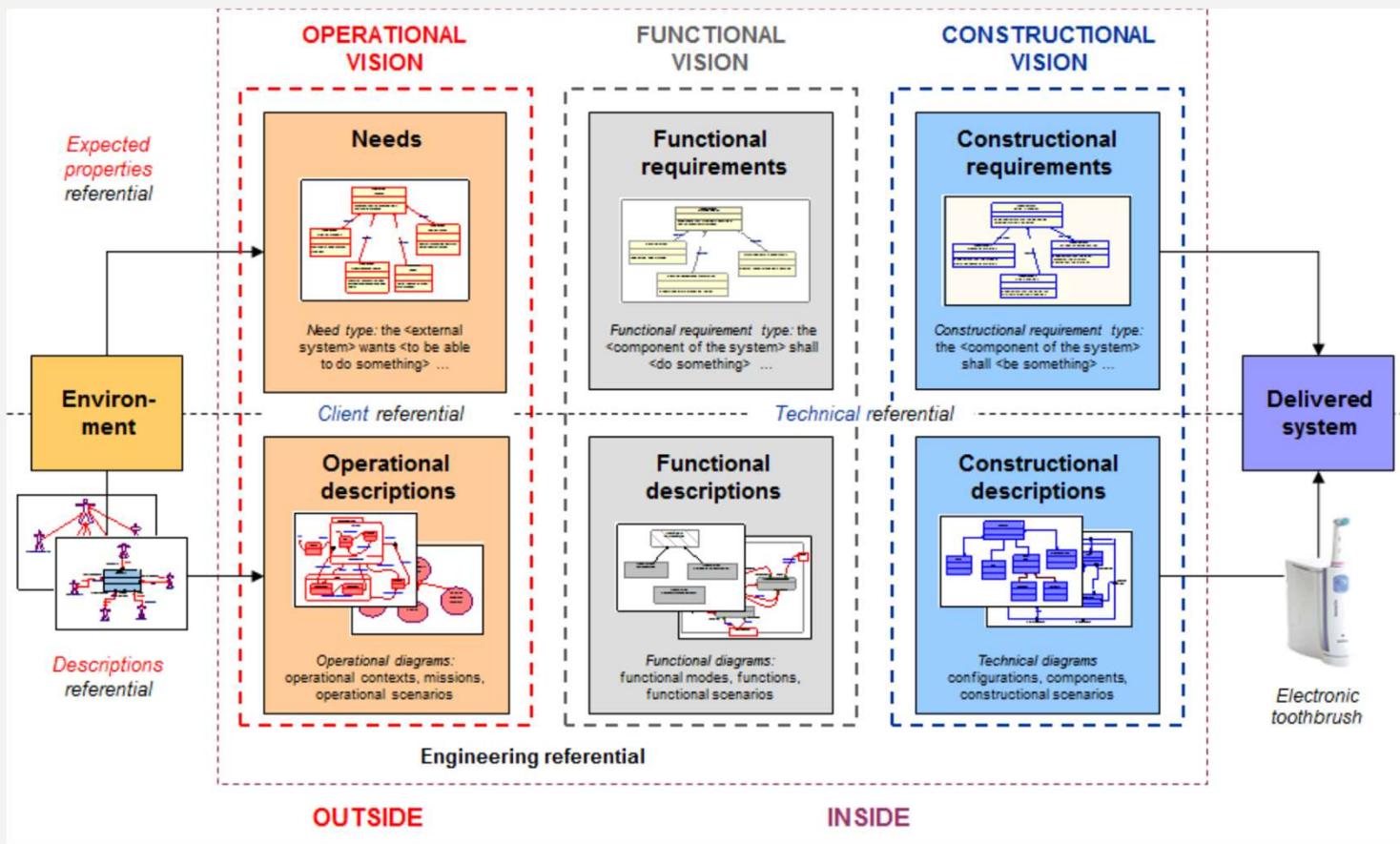
# SYSTEM ARCHITECTURE MATRIX (CESAM)

Visions	Expected properties	Descriptions			
		States	Static elements	Dynamics	Flows
<i>Operational vision</i>	Needs	Operational contexts	Missions <sup>124</sup>	Operational scenarios	Operational flows or objects
<i>Functional vision</i>	Functional requirements	Functional modes	Functions <sup>125</sup>	Functional scenarios	Functional flows or objects
<i>Constructional vision</i>	Constructional requirements	Configurations	Components <sup>126</sup>	Constructional scenarios	Constructional flows or objects

# Matrice 9 vues(CESAM)

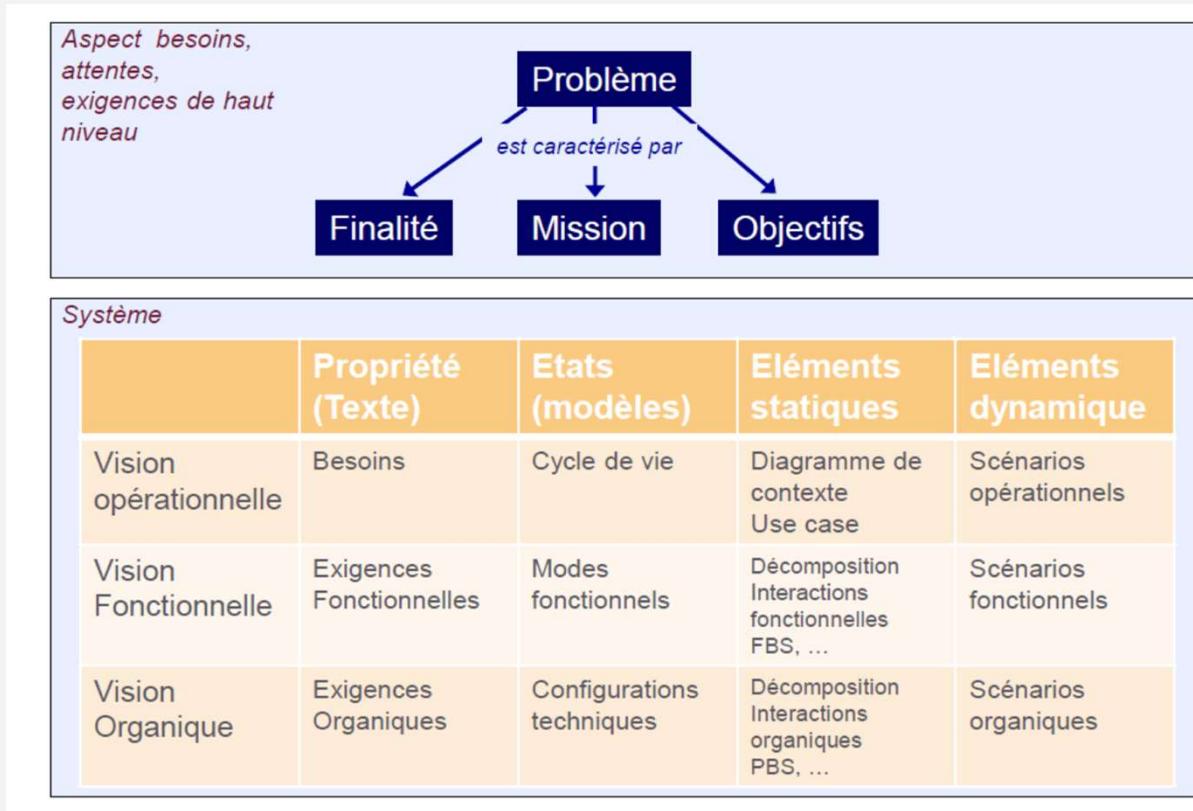


# La matrice d'architecture systèmes de CESAM



# MÉTHODE D'INTERVENTION (PFMOS)

- PFMOS : *Problème -Finalités –Missions –Objectifs -> Système*



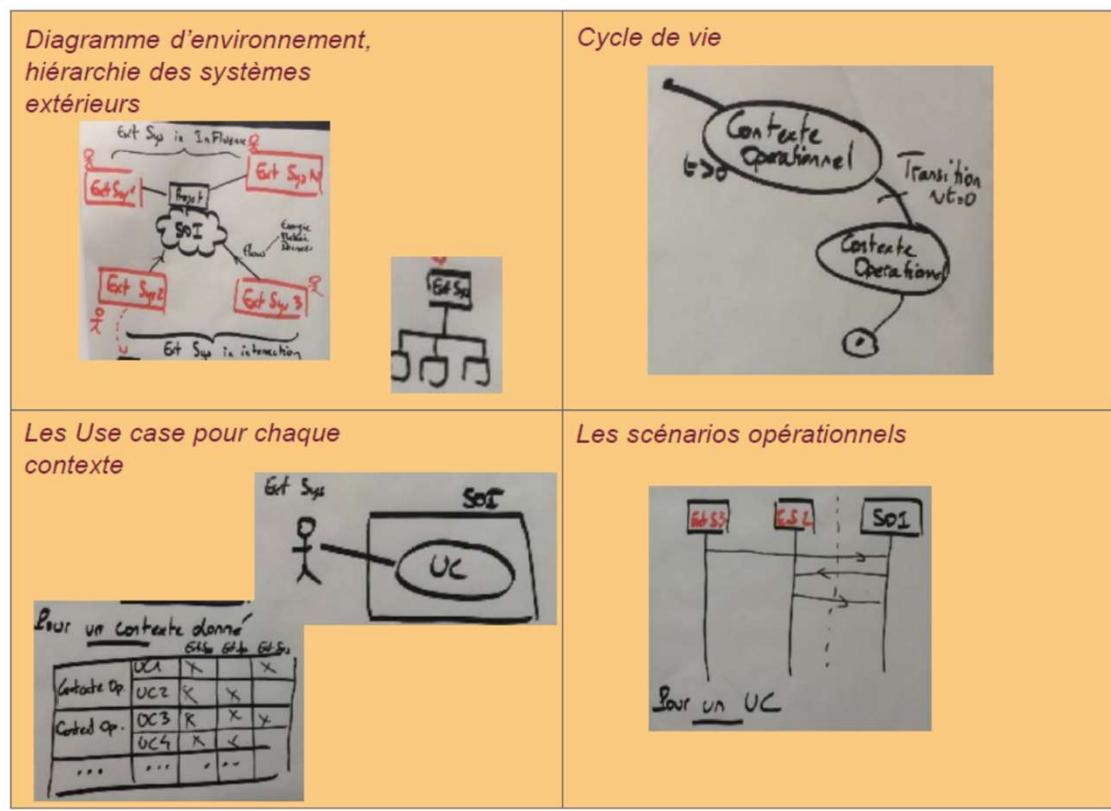
*avant d'aller plus loin, voila quelques définitions ...*

- **Besoins fonctionnels** = ce que doit FAIRE le système pour répondre à sa finalité
- **Missions** = ce que le système doit accomplir dans les différents contextes d'opération  
« quoi »
- **Finalité** = une raison d'être « pourquoi? » qui justifie la décision de réaliser un système relativement à la résolution d'un problème

- Le *concept d'opération* décrit de façon macroscopique ce que doit faire le système et pourquoi + Performances et **objectifs clés** à atteindre
- **Scénarios opérationnels et services** : Une mission s'analyse comme un enchaînement de *services* rendus à l'environnement par le biais de *fonctions*
- **Exigence** = énoncé prescrivant une fonction, une aptitude, une caractéristique, une limitation à laquelle doit satisfaire un produit
- **Les « parties prenantes »** = Personnes physiques et organisations concernées directement ou indirectement par le futur système

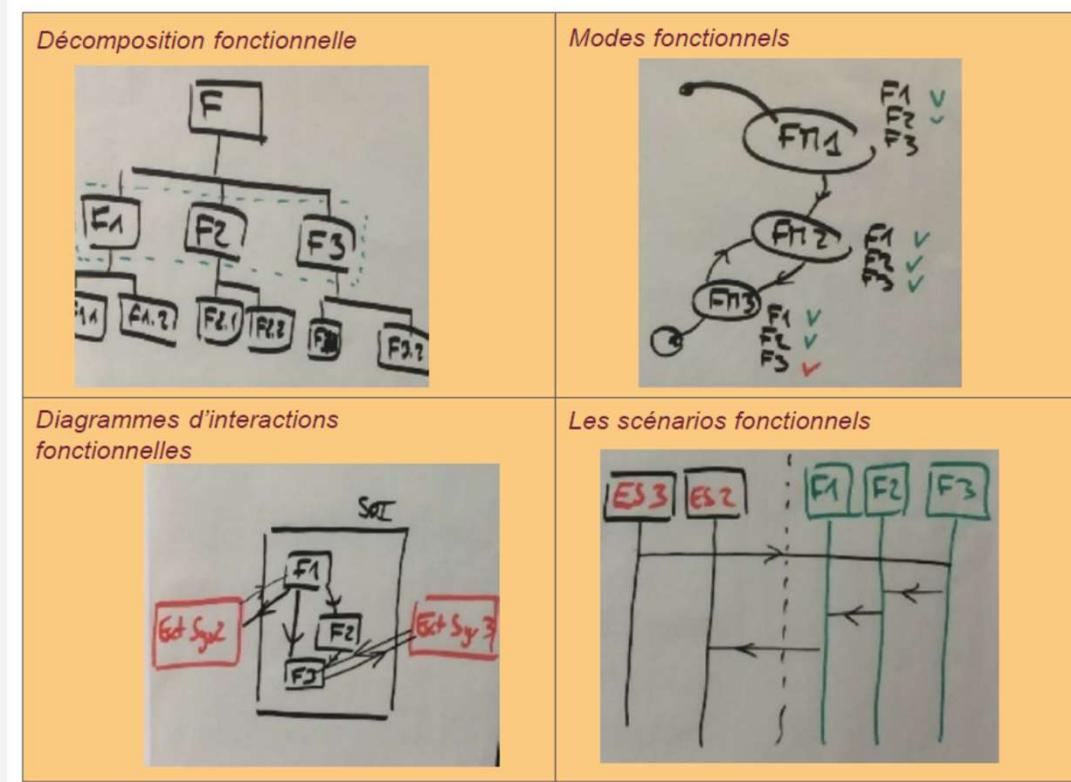
# Méthode d'intervention (PFMOS)

## Vue opérationnelle -> why



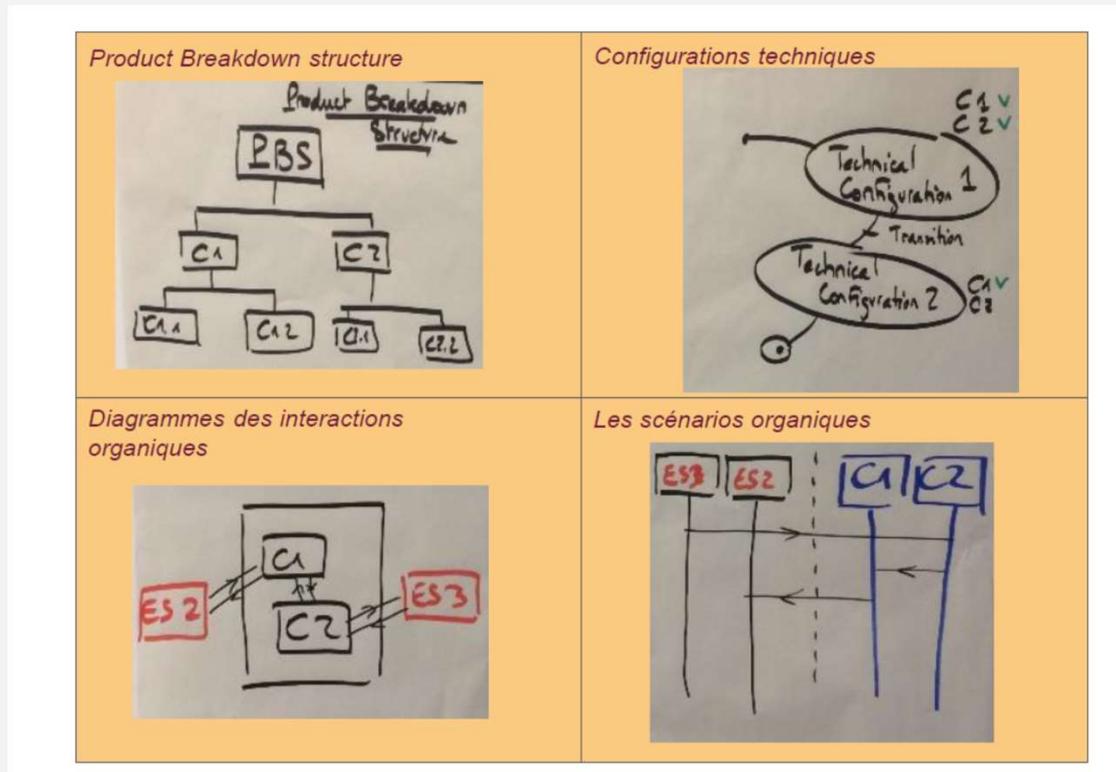
# Méthode d'intervention (PFMOS)

## Vue Fonctionnelle -> WHAT)



# Méthode d'intervention (PFMOS)

## Vue Organique -> HOW

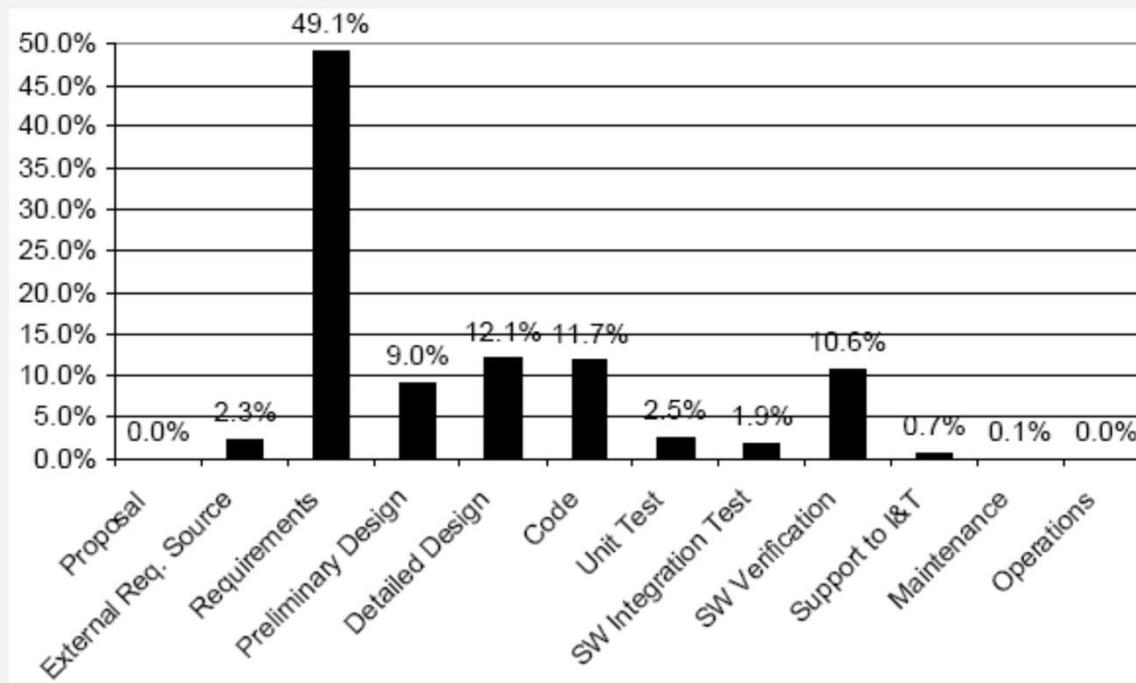


# **CHAPITRE 3 : INGENIERIE DES EXIGENCES**

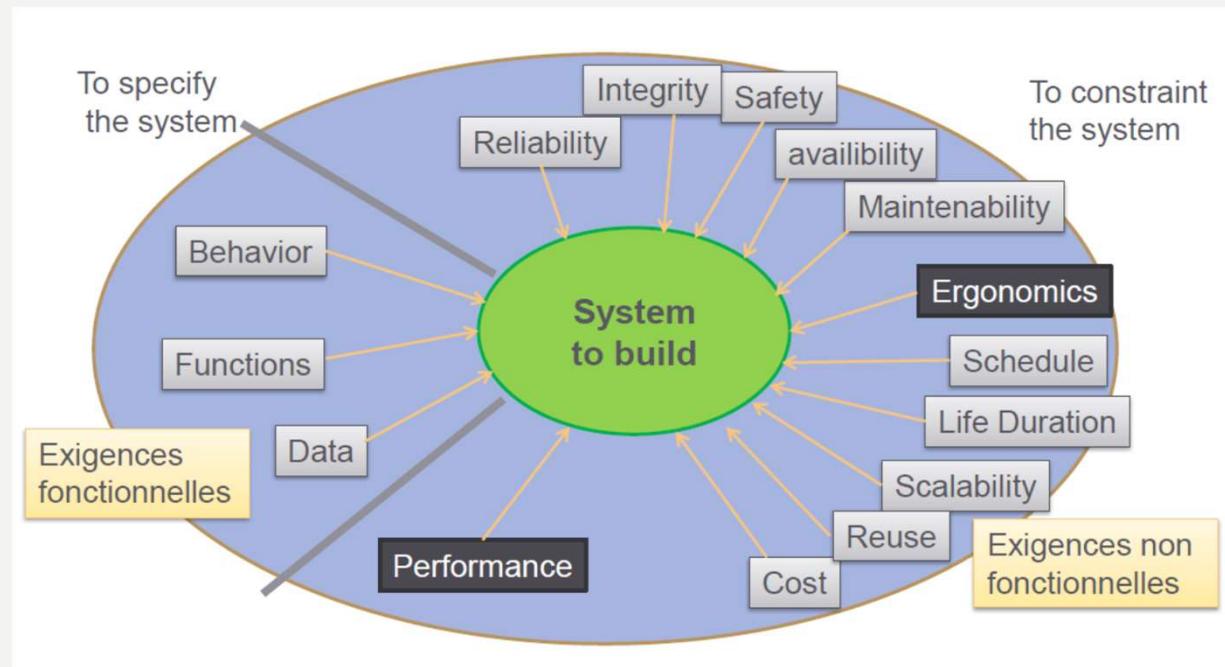
- *La phase de recueil des exigences est une étape clé pour la réussite de tout projet. Elle vise à faire une traduction/représentation des besoins du client en un formalisme plus précis qu'un cahier des charges par exemple*

# RÔLE DES EXIGENCES DANS LES PROCESS IS

- Plusieurs études ont montré sur les faiblesses dans la définition des besoins et des exigences constituent la cause majeure des échecs des projets

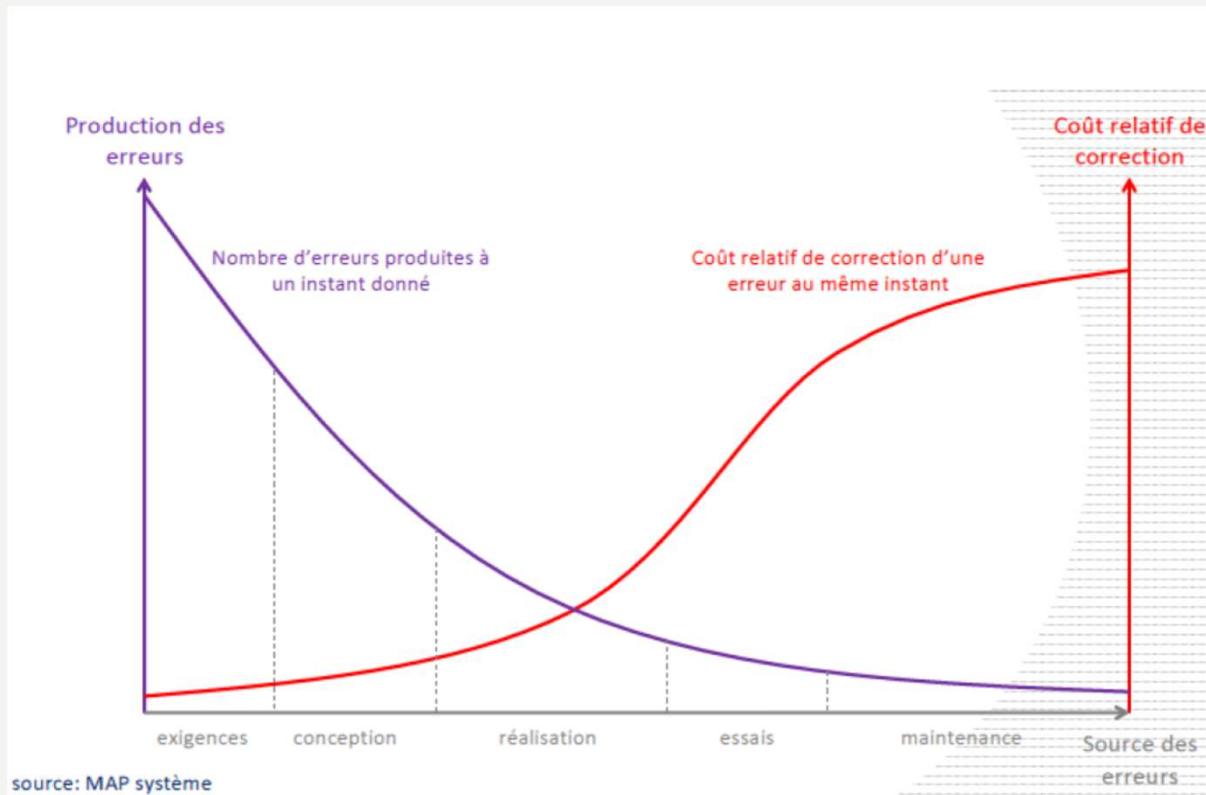


# DÉFINITION DES EXIGENCES

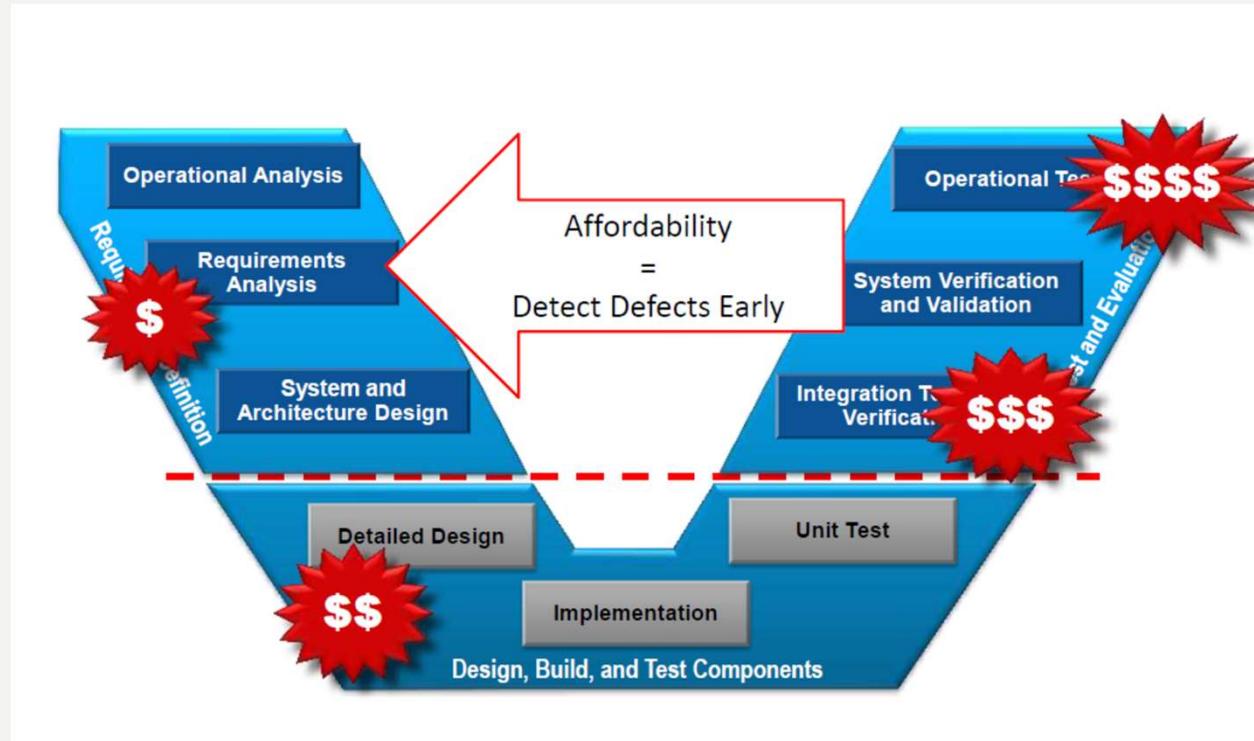


- **Functional requirements = What the system must do**
- **Non-functional requirements = Qualities the system must have**

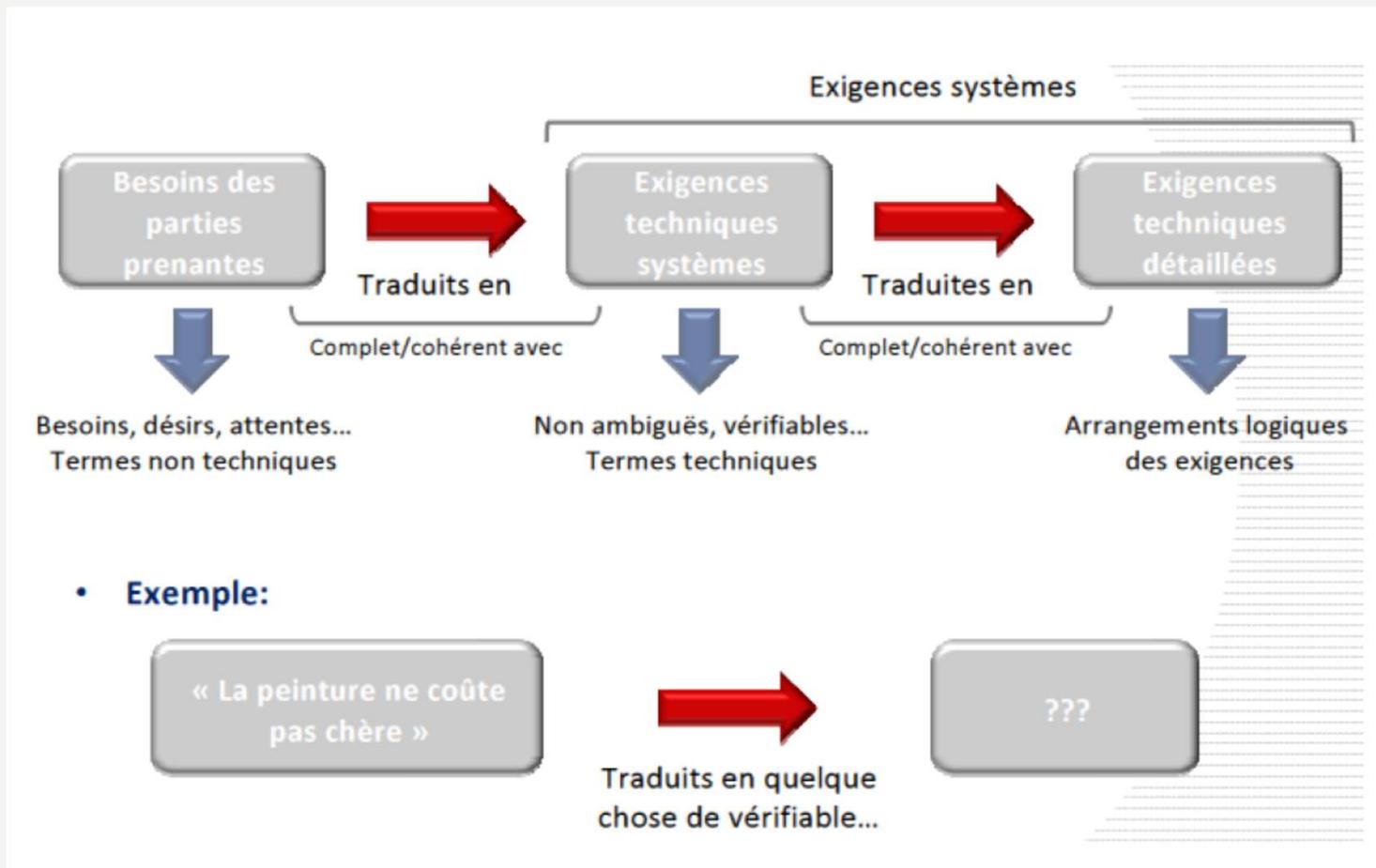
# ORIGINE DES ERREURS ET COÛT DE CORRECTION



# Detect defects earlier



# DES BESOINS AUX EXIGENCES



# ELICITATION DES EXIGENCES

## Problèmes dans les spécifications



- Exigences ambiguës
- Exigences non spécifiées au bon niveau de granularité
- Exigences non vérifiables et non mesurables
- Spécifications trop détaillées

## Elicitation des exigences



- Des revues d'exigences entre le client et le fournisseur
- Clarification des significations des exigences
- Engagement sur les responsabilités et sur les processus de validation
- Mettre l'exigences au bon niveau

# BONNES PRATIQUES POUR ÉCRIRE LES EXIGENCES

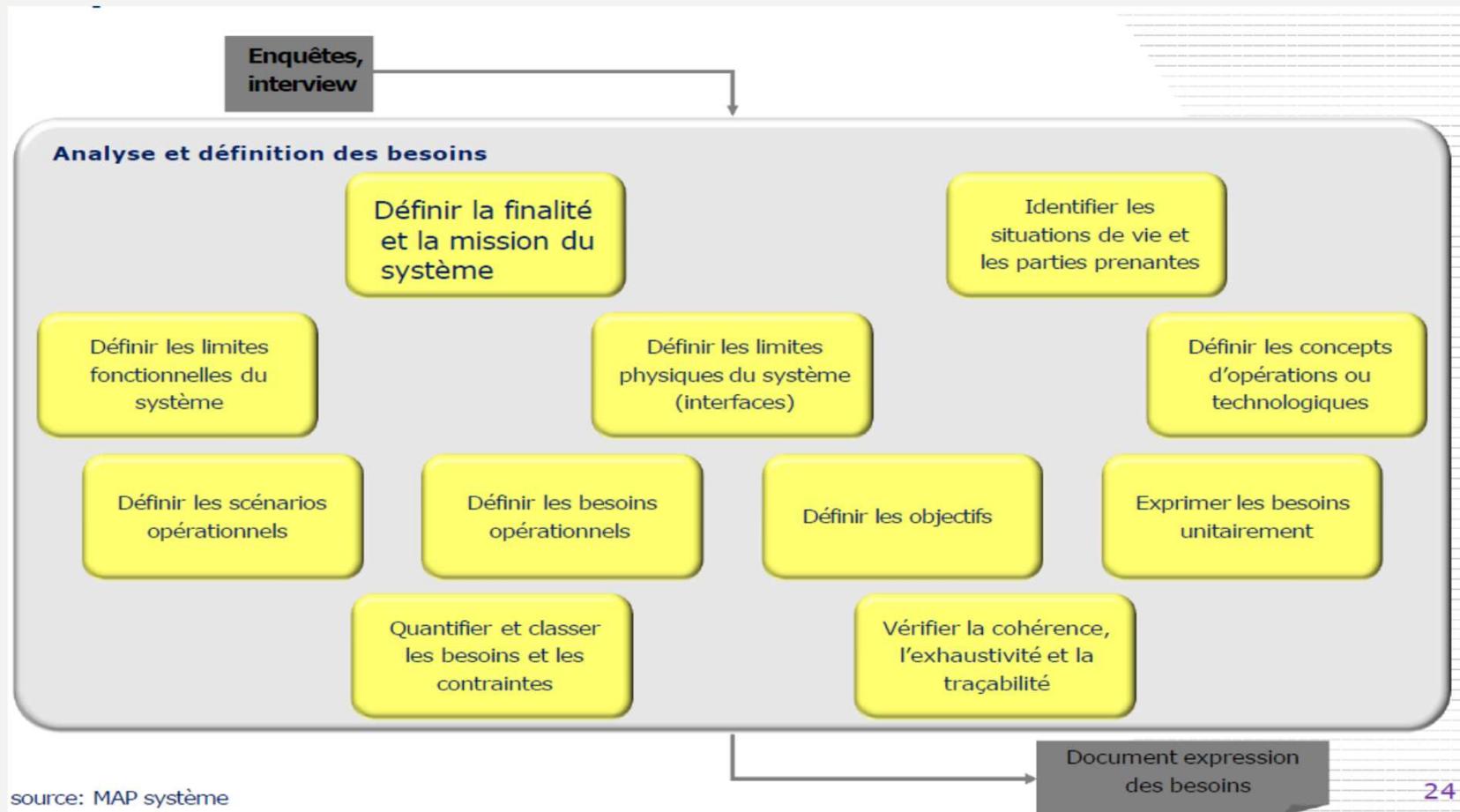
Chaque exigence doit être **SMART**

- > **S**pecific
- > **M**easurable
- > **A**chievable
- > **R**elevant
- > **T**raceable

## Requirement

- Identifier : <ident>
- Description : <Text>
- Verification type : test/observation/...
- Classification : functional, performance, constraint, quality, ...
- Relevance : Mandatory, strong, week, optional
- Maturity : origin, analyzed, approved, verified, ...

# LE PROCESSUS DE DÉFINITION DES BESOINS DES PARTIES PRENANTES



source: MAP système

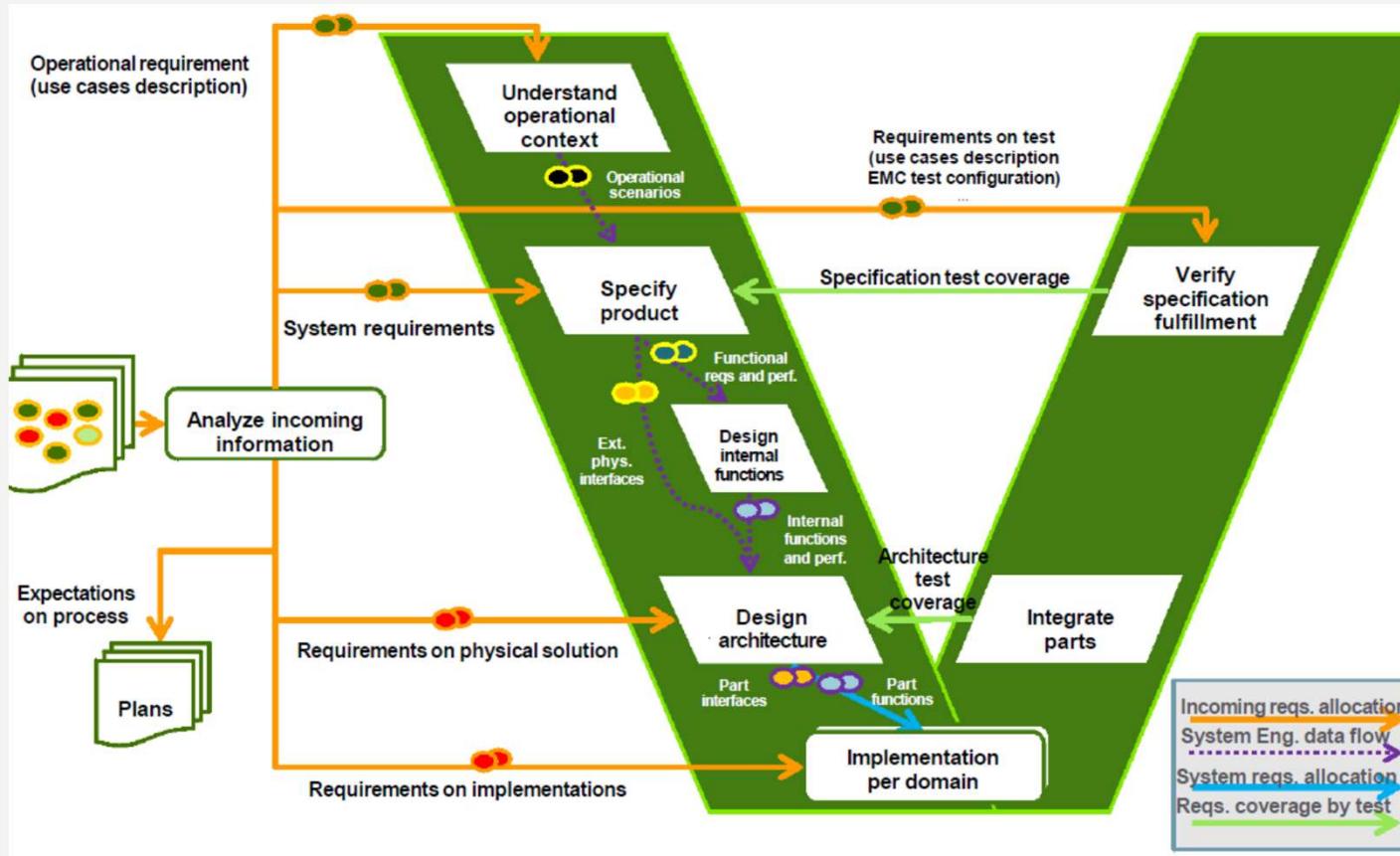
30/03/2020

N.MACHKOUR

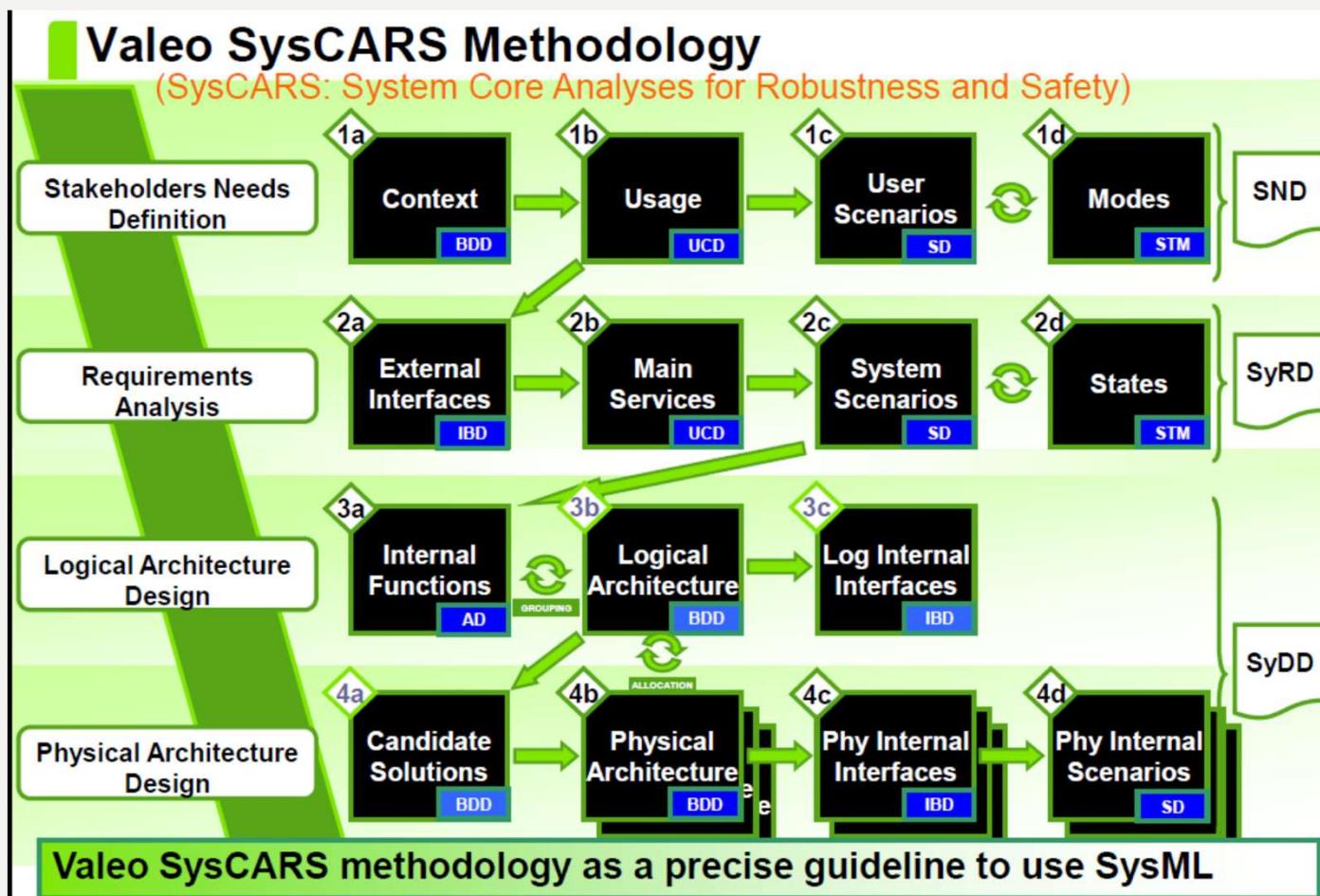
24

89

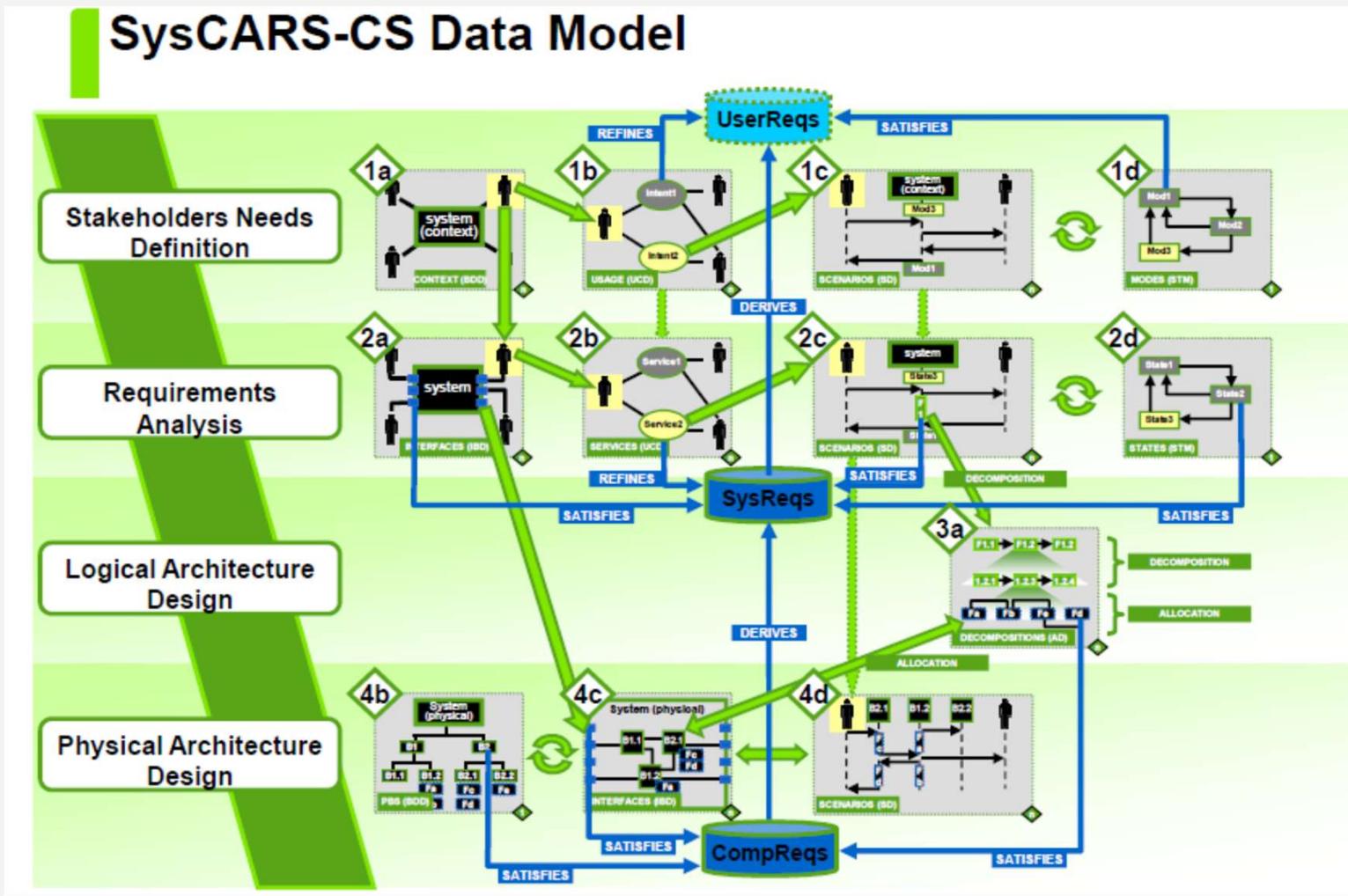
# REQUIREMENTS FLOWS IN THE PROCESSES



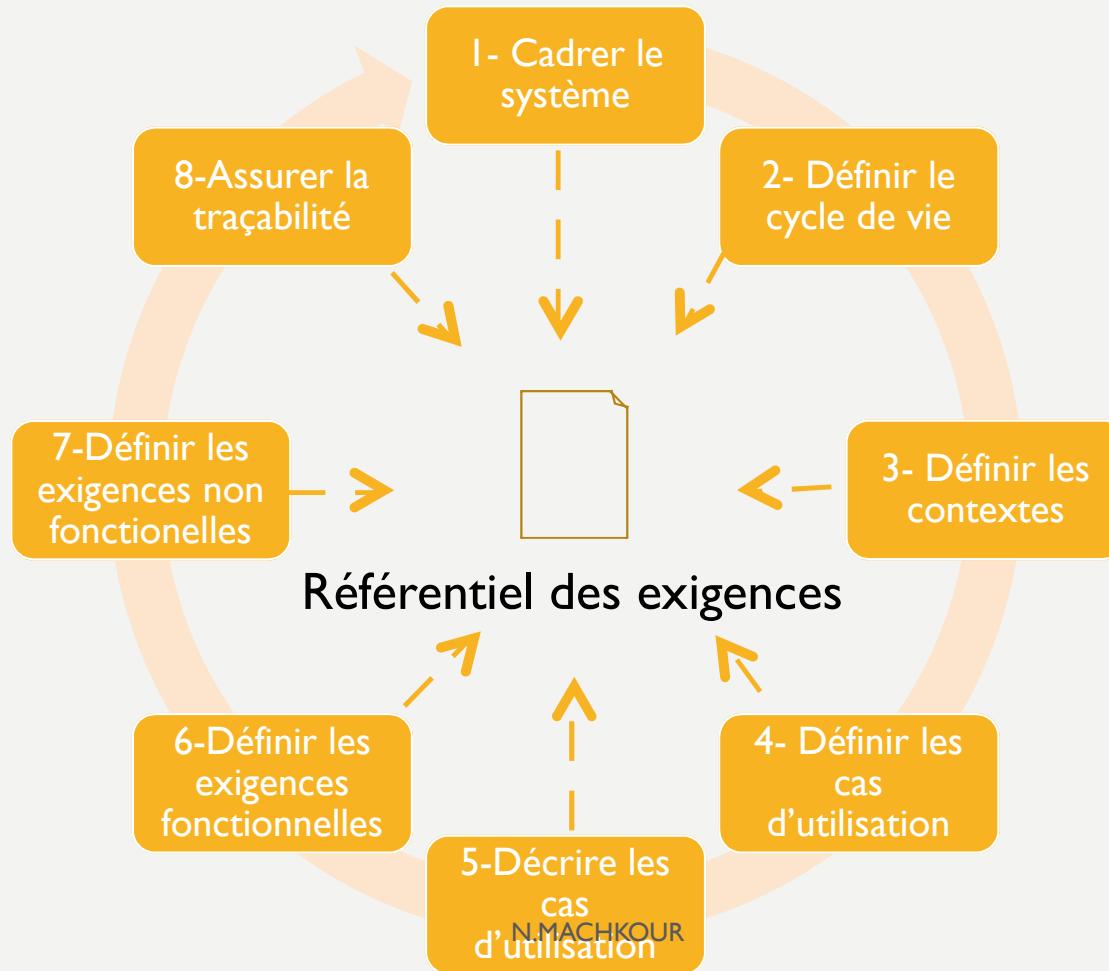
# Benchmark VALEO



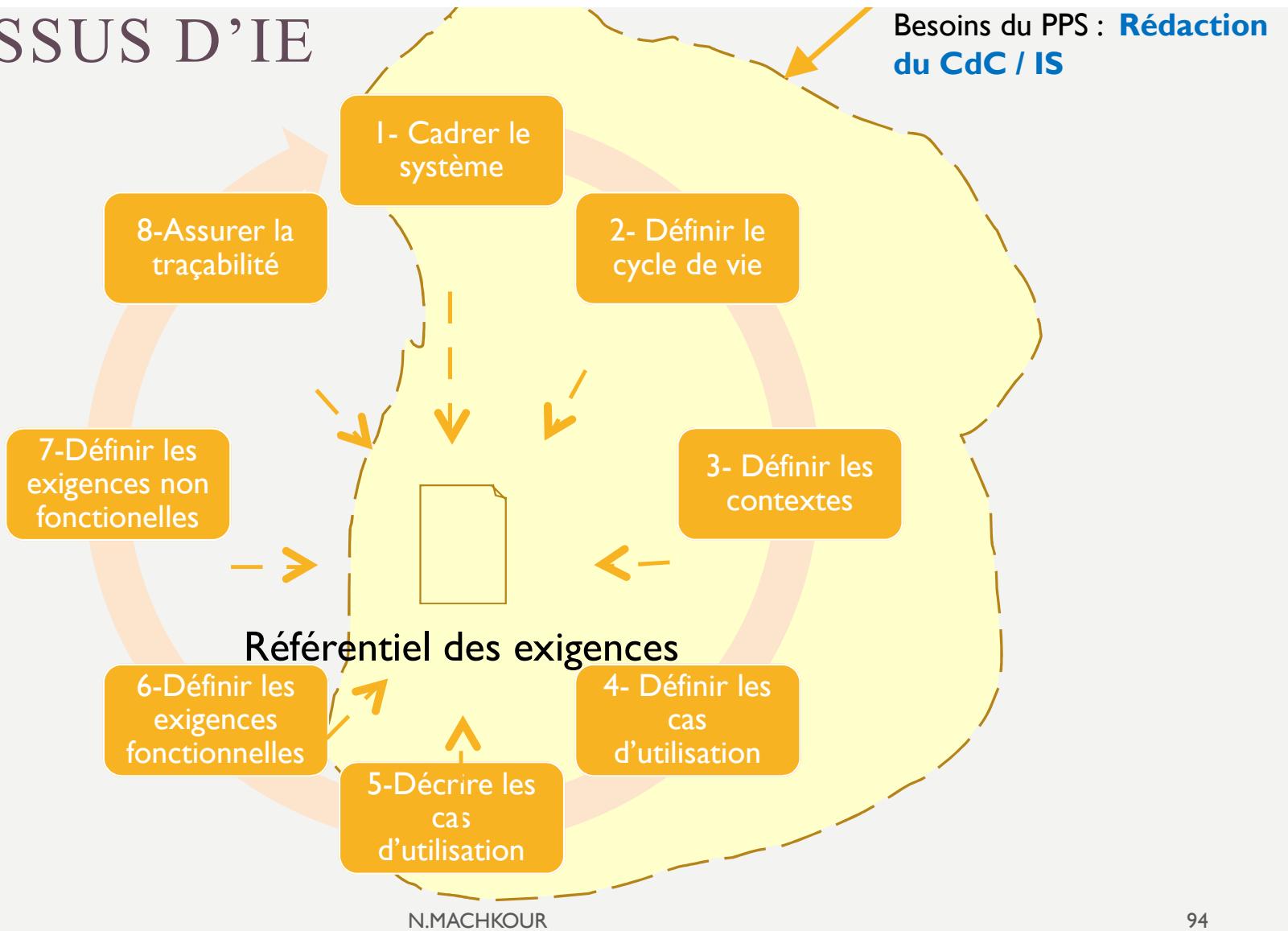
# Gestion des Exigences chez VALEO



# PROCESSUS D'IE

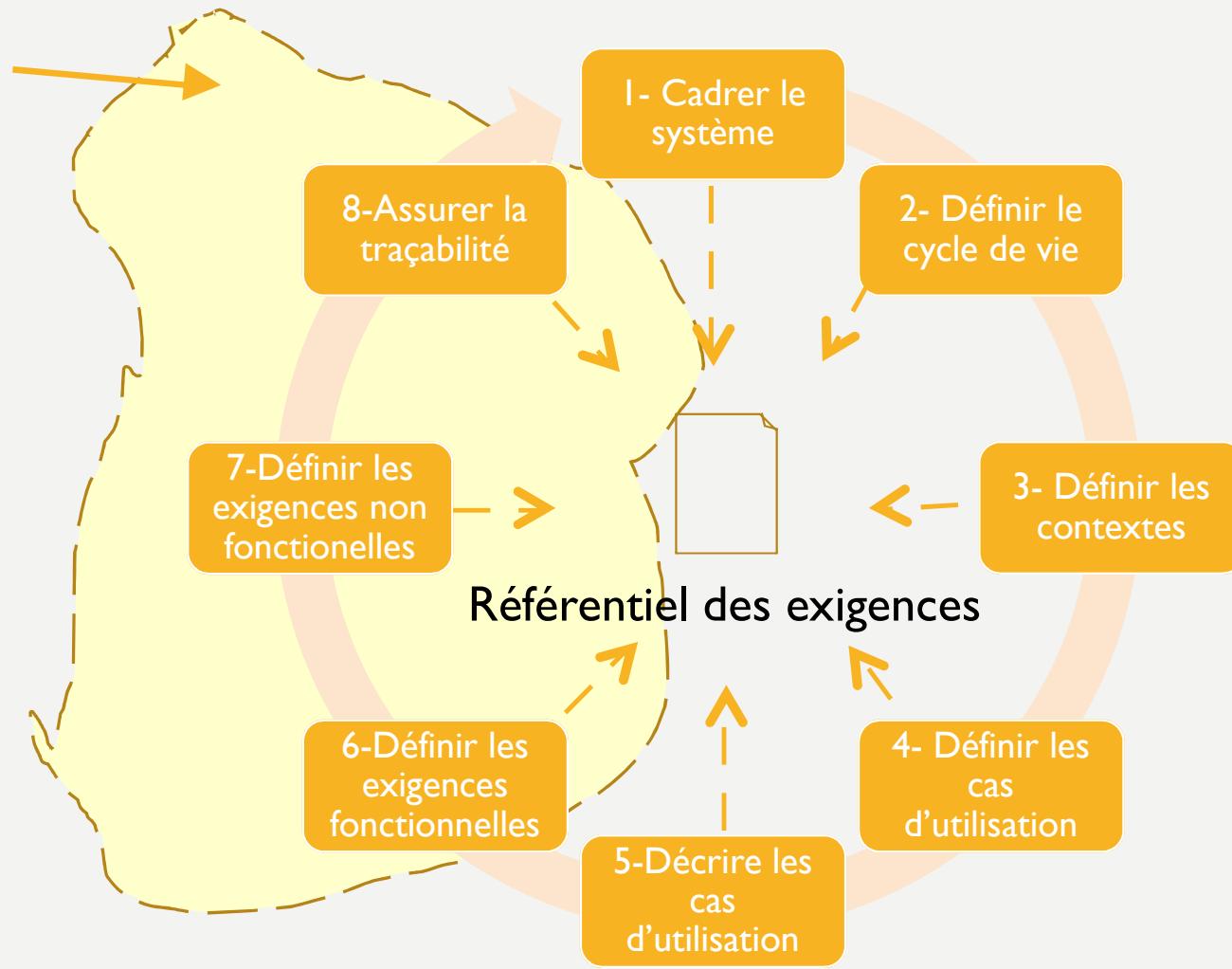


# PROCESSUS D'IE

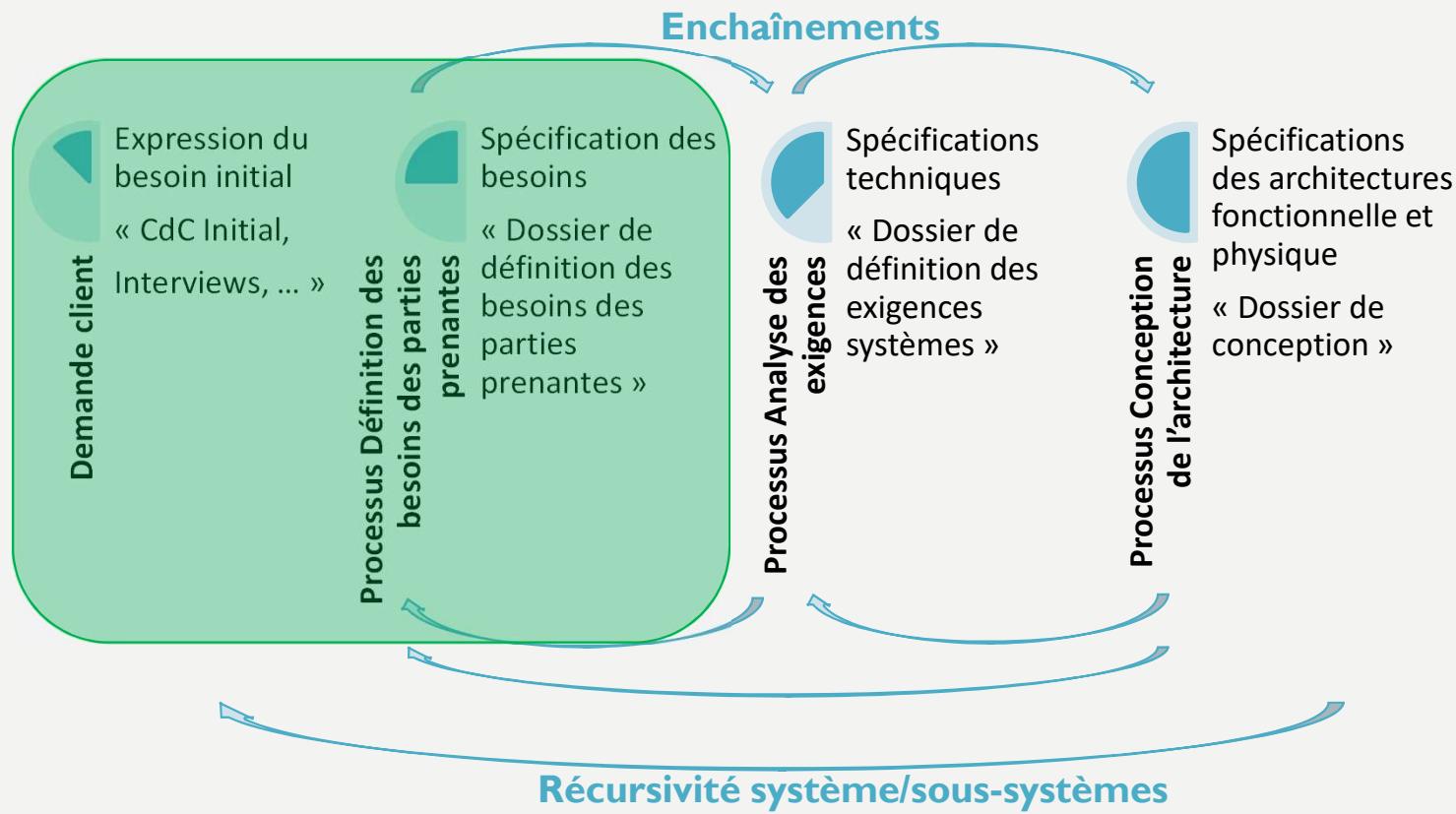


# PROCESSUS D'IE

Sous Processus :  
Analyse des exigences



# DÉFINITION DES BESOINS DES PPS RÉDACTION DU CDC / IS

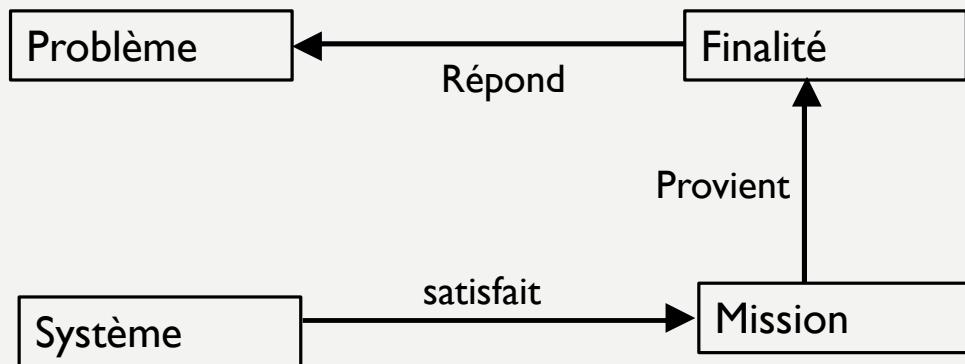


# 1-Cadrer le système

Une première analyse du besoin doit être menée pour définir la mission principale du système. Cette première analyse cadre globalement le système à faire.

Pourquoi on veut faire ça ? → finalité

Qu'est-ce que l'on doit faire ? → mission



La mission peut être raffinée :

- Pour spécifier des sous missions qui dérivent de la mission principale
- Pour spécifier des performances, contraintes, ...

## 2-définir le cycle de vie

Définir le Cycle de vie pour :

- Identifier les contextes relatifs à ses différentes phases de vie (exploitation ou utilisation, conception, réalisation ou production, soutien, retrait ou fin de vie, ...).
- Identifier les PPS Pour chaque phase du cycle de vie, (Quelles sont les parties prenantes pouvant exprimer des contraintes ou des besoins quand le système est dans cette phase du cycle de vie ?)



## 3-Définir Les contextes

Pour chaque phase du cycle de vie où des services sont attendus du système, on définit un diagramme de contexte du système.



### Objectifs :

Identifier les parties prenantes ;

Identifier les éléments externes en interaction avec le système ;

Définir les frontières du système et de son contexte.

Selon [AFIS], les parties prenantes sont toutes les personnes (physiques ou morales) concernées directement ou indirectement par le système dans toutes ses situations de vie :

- Côté acquéreur – Intéressées par l'utilisation : Organisme acquéreur, utilisateurs, exploitants, opérateurs, mainteneurs, ...
- Côté fournisseur – Impliquées dans la réalisation : Concepteur, réalisateur, mainteneurs, sous-traitant, ...
- Potentiellement concernées : Impactées plus ou moins directement et à plus ou moins long terme comme la société, des associations, ...
- Organismes de certification, homologation, qualification, ...

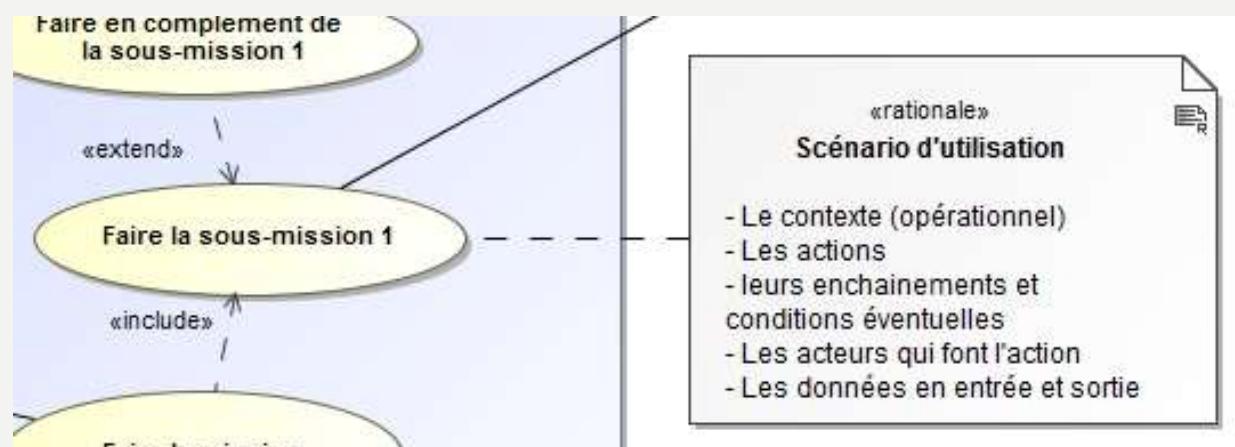
## 4-Définition des utilisations

- Pour chaque phase du cycle de vie où des services et des fonctionnalités sont attendus du système, on définit les cas d'utilisation du système.
- Certaines phases de vie ne donnent pas toujours lieu à la mise en place de fonctionnalités rendue par le système (conception, maintenance)
- Généralement la mission principale, se retrouve dans le cas d'utilisation principal de la phase d'exploitation
- utilisation = besoin de service attendu.



## 5-Définition des scénarios

- Pour chaque cas d'utilisation, on définit un scénario d'utilisation nominale de manière textuelle :



# Définition des besoins des Parties Prenantes

- A partir des éléments initiaux : mission, finalité, besoins, contraintes, complétés sur la base des analyses précédentes : étude des services attendus, étude du contexte, étude des scénarios.
- Ceux-ci sont classés de la façon suivante :
  - Service attendu ;
  - Opérationnel (mode de fonctionnement, modes de marche, condition d'évolution, ...) ;
  - Performance ;
  - Interface (physique, ergonomie, interopérabilité, ...) ;
  - Contrainte (liée à une phase de vie, environnement du système, réglementation, coût, délai, ...).

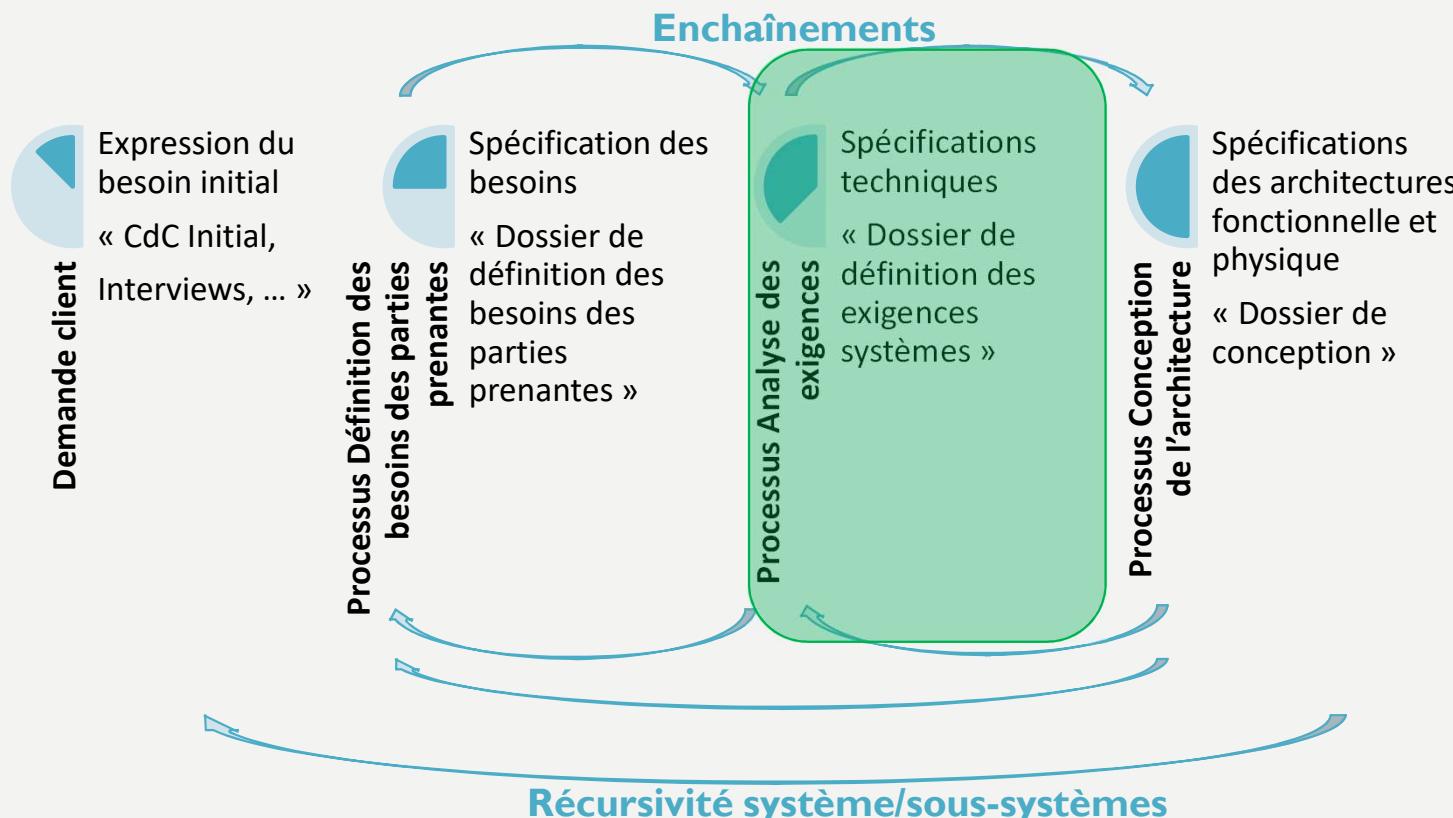
# En résumé

La spécification des besoins permet donc de répondre à :

- Pourquoi on veut faire ça ? → finalité
- Qu'est-ce que l'on doit faire ? → mission
- Qui est concerné / impacté ? → parties prenantes
- Quelles sont les frontières du système ? → contexte
- Quels services sont attendus ? → utilisations
- Comment cela s'envisage t-il ? → scénarios
- Quels sont mes besoins pour répondre à tout cela ? → besoins

L'ensemble de tous les diagrammes obtenus durant ce processus  
**constitue le cahier des charges.**

# ANALYSE DES EXIGENCES



# ANALYSE DES EXIGENCES

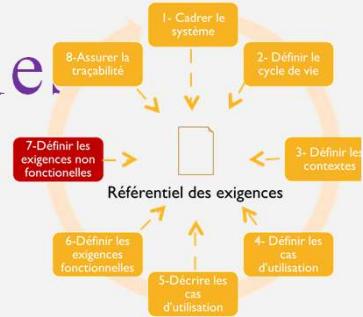
- Sur la base des besoins des parties prenantes, ce processus technique englobe :
  - Apport des concepts systèmes ;
  - Description des états initiaux (SMD), raffinés par la suite ;
  - Description précise des scénarios (SD) ;
  - Définition des exigences système (RD).
- Celles-ci sont classées de la façon suivante :
  - Fonctionnelle ;
  - Opérationnelle (mode de fonctionnement, modes de marche, condition d'évolution, ...) ;
  - Performance ;
  - Interface (physique, ergonomie, interopérabilité, ...) ;
  - Contrainte (liée à une phase de vie, environnement du système, réglementation, coût, délai, ...) ;
  - Validation (Tests ou essais, inspections, revues ou audits, ...).

## 6- Définir les exigences fonctionnelles



- Chaque utilisation (fonctionnalité) définie précédemment donne lieu une exigence fonctionnelle :
  - à minima un nom et un identifiant
  - On peut rajouter un type, une description textuelle
- Le nom et/ou la description textuelle doivent présenter des caractéristiques de qualité et sont souvent rédigés en utilisant un gabarit (voir annexe du document support de TP)
- On utilise souvent un tableau pour les recenser.

## 7- Définir les exigences Non fonctionnelles



- Pour chaque cas d'utilisation et sa description, avec un questionnement systématique relatif aux types d'exigences prédéfinis : « Quelles sont les performances attendues ? », « quelles sont les interfaces permettant l'interaction ? », ...
- Pour chaque contexte et chaque PPs, avec un questionnement relatifs aux contraintes inhérentes : « Quelles sont les contraintes de la PPs dans ce contexte
- Utiliser les mêmes règles que précédemment (exigences fonctionnelles) : (nom, identifiant,...)

## 8- Assurer la traçabilité



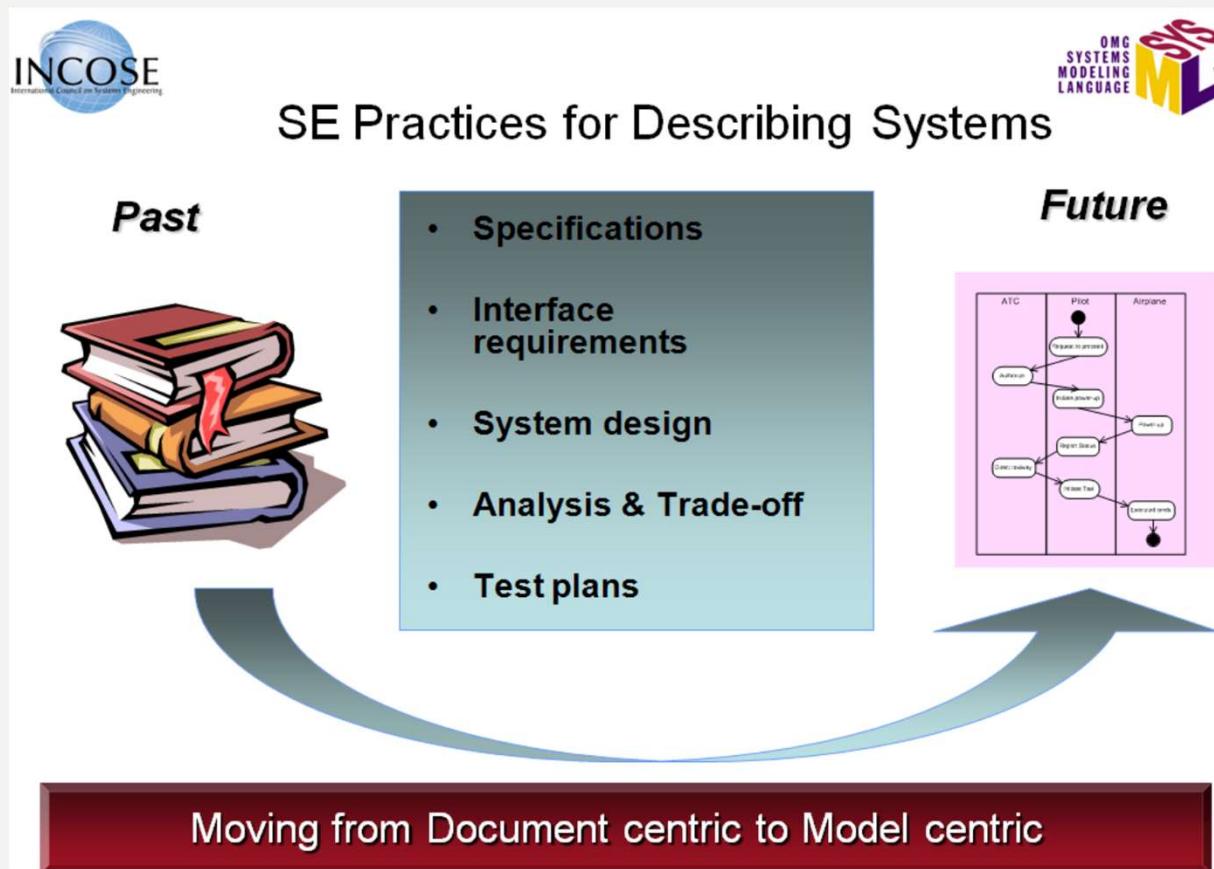
- Assurer la traçabilité des exigences entre les processus d'ingénierie des exigences
  - Définition du besoin, analyse des exigences, exigences des constituantes
- Vérifier la traçabilité des exigences :
  - Toutes les exigences d'un niveau sont prises en compte : exemple besoin des PPs devraient l'être )
  - Aucune exigence n'est orpheline (superflue) : exemple d'une exigence qui serait ni liée à un besoin ni dérivée d'une autre exigence et qui risquerait de sur-constrainer le système

# **CHAPITRE 4 : MBSE(MODEL BASED SYSTEM ENGINEERING**

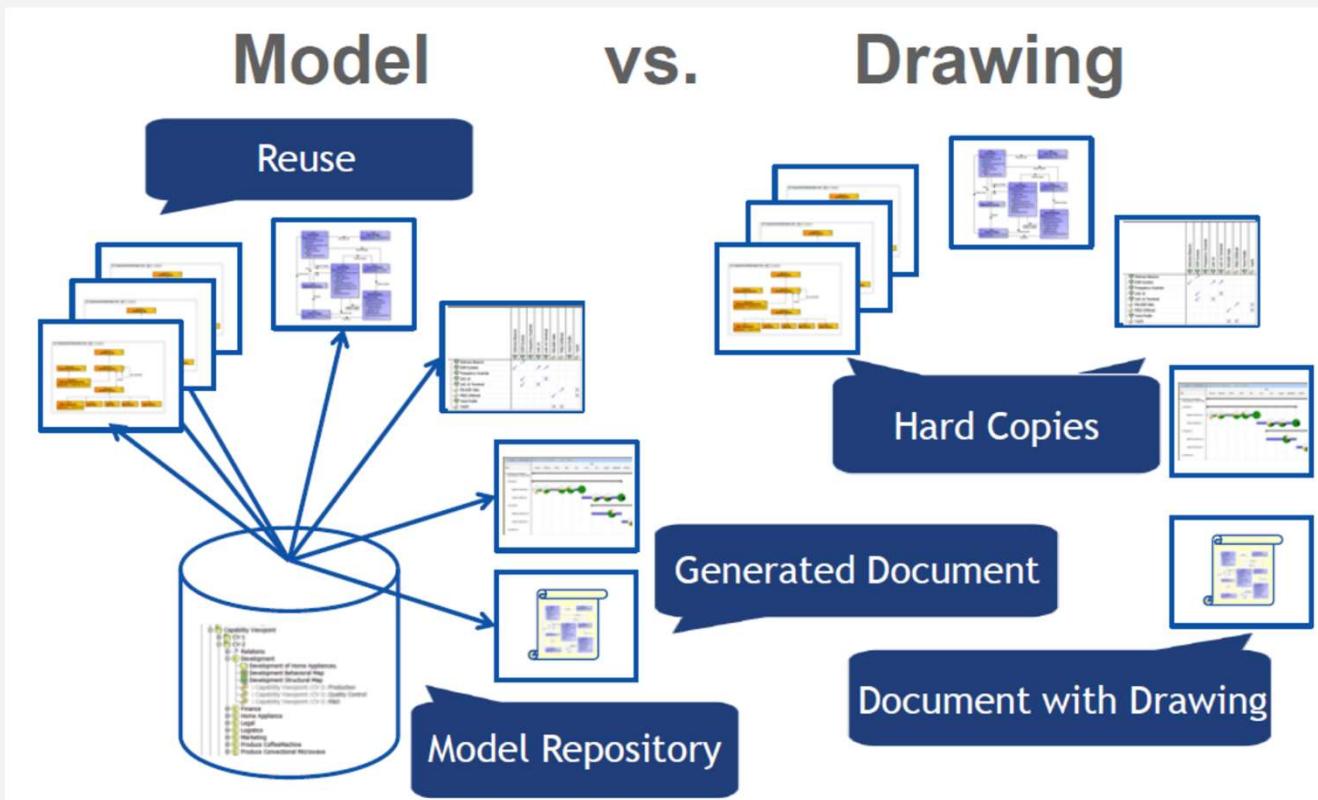
# MBSE

- C'est l'Ingénierie des systèmes utilisant des modèles,
- C'est l'application formalisée de la modélisation pour soutenir :
  - ✓ Exigences du système
  - ✓ Analyse
  - ✓ Conception
  - ✓ Activités V&V (Vérification, Validation)

# DOCUMENT-BASED SE VS MBSE



# MODEL VS. DRAWING



## *Avantages des modèles et de la modélisation*

- ✓ Gérer la complexité
- ✓ Améliorer la qualité
- ✓ Améliorer la communication
- ✓ Permettre une collaboration inter fonctionnelle plus étroite
- ✓ Préserver la connaissance
- ✓ Réutiliser

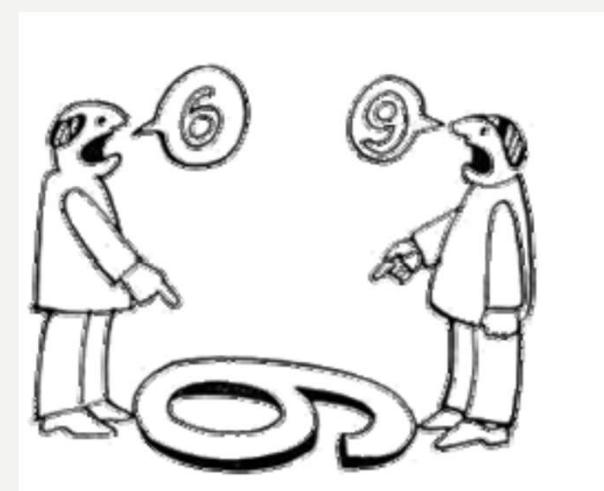
- **Modeling solution is a combination of a modeling language(s), a methodology and a modeling tool that together provide a productive infrastructure for applying model-driven development in the context of a particular organization.**

## Remarque

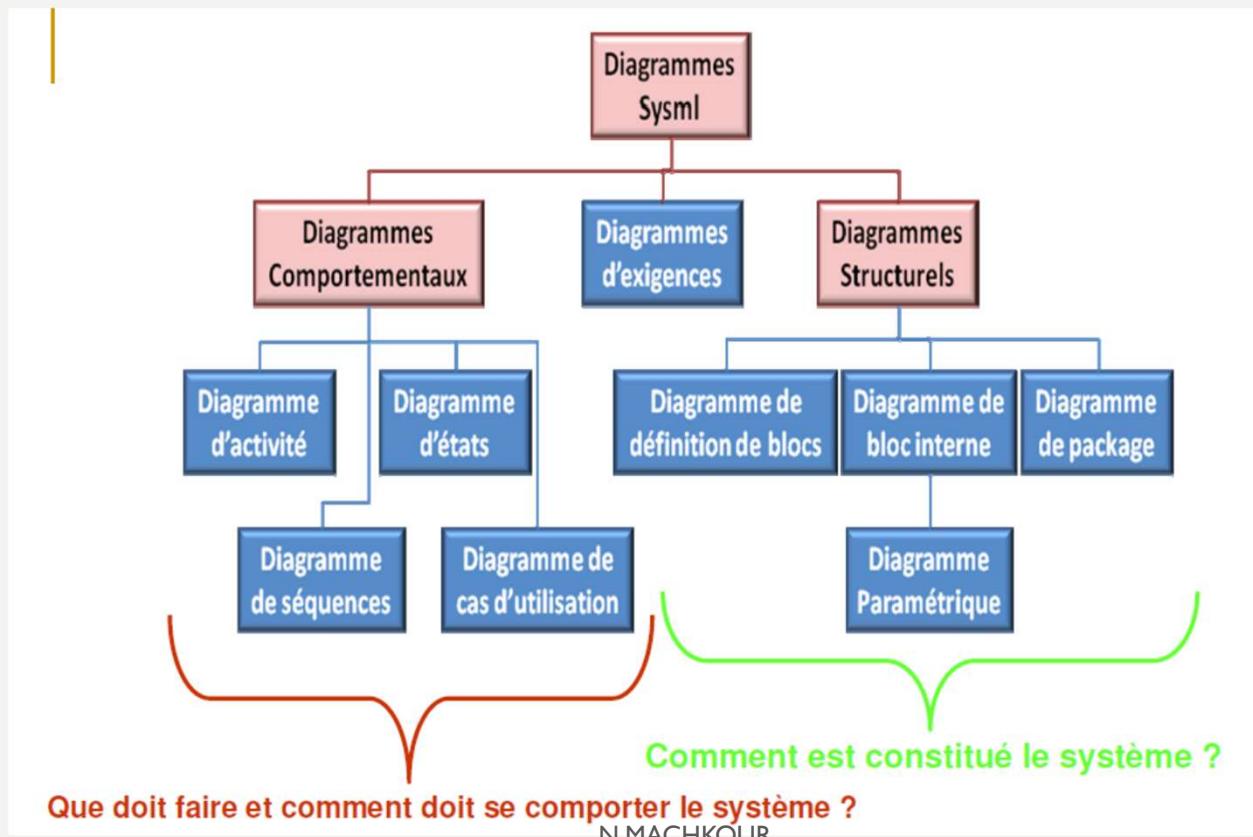
- The modeling language is just the language, and must be combined with a methodology to be useful

# SYSTEMS MODELING LANGUAGE (SYSML)

- Le langage SysML est un langage de **modélisation** permettant de **décrire** tout ou partie d'un système technique, d'un point de vue transversal, comportemental ou structurel
- C'est un outil de **communication** utilisé dans toutes les phases de vie du produit afin que les différents acteurs puissent **communiquer et mesurer les écarts** entre le produit en cours de réalisation et celui souhaité
- N'est pas une méthode d'étude, de réflexion ou de conception



*Il s'articule autour de 9 types de diagrammes*



- Dans ce cours on va se limiter à l'utilisation de 6 diagrammes : Diagramme d'utilisation, diagramme d'état, diagramme de séquence, diagramme d'exigences, diagramme de définition de blocs, et diagramme des blocs internes
- **Indicateurs des 6 diagrammes SysML**

<b>uc</b> : diagramme des cas d'utilisation	<b>stm</b> : diagramme (machine) d'état
<b>req</b> : diagramme des exigences	<b>bdd</b> : diagramme de définition des blocs
<b>sd</b> : diagramme de séquence	<b>ibd</b> : diagramme des blocs internes

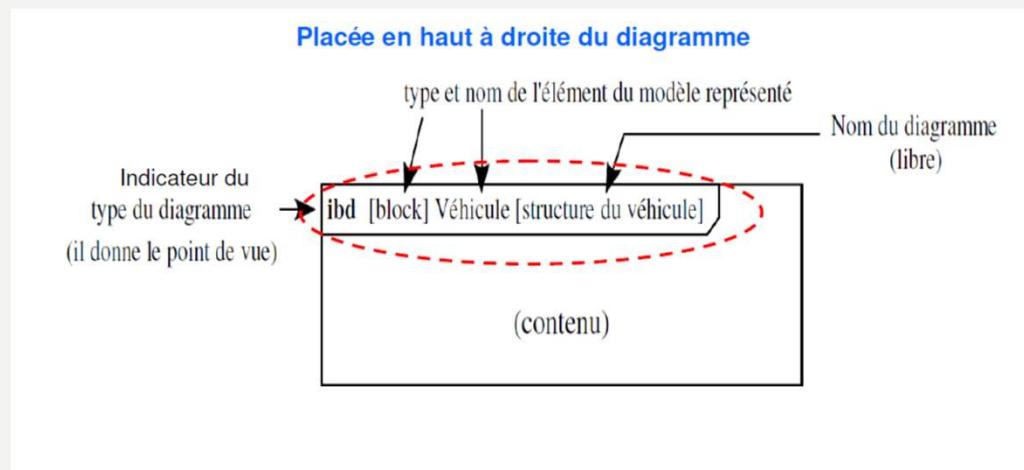
## Remarque

- Aucun ordre d'analyse des **6** diagrammes n'est préconisé, un travail simultané sur l'ensemble est souhaité
- Toute modification de l'un des diagrammes engendre généralement des modifications dans les autres représentations

# Principales relations utilisées et leurs significations :

Relation	Désignation	Signification	Utilisée dans
A → B	Association	relation d'égal à égal entre deux éléments. A utilise B	uc bdd ibd
----->	Dépendance	2 items distincts mais dont l'un dépend de l'autre A dépend de B	uc req bdd
—◇—	Agrégation	un élément est une composante facultative de l'autre. A entre dans la composition de B sans être indispensable à son fonctionnement	req bdd
—◆—	Composition	un élément est une composante obligatoire de l'autre. A entre dans la composition de B et lui est indispensable	req bdd
—►—	Généralisation	dépendance de type « filiation » entre 2 items A est une sorte de B	uc bdd ibd
—⊕—	Contenance	relation d'inclusion entre 2 items B contient A	uc req bdd

# Cartouche de repérage



## Remarque : Need for a Method/Framework

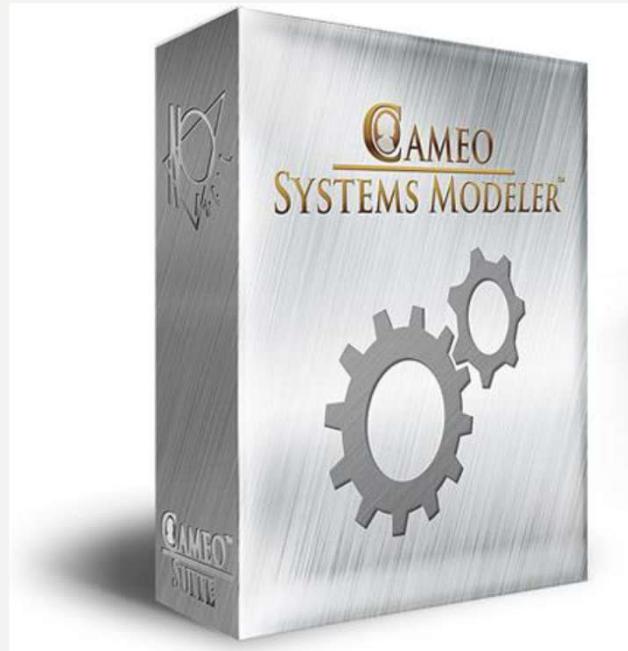
- **This opens discussions of:**

- how to structure the model
- what views to build
- which artifacts to deliver
- and in what sequence

**Every company deals with the same issues differently. Some use:**

- defense architecture frameworks: DoDAF, NAF, **MODAF**
- MBSE methods: **OOSEM, Harmony, SYSMOD, FAS**; however, saying there is no need for an architectural framework just doesn't work.

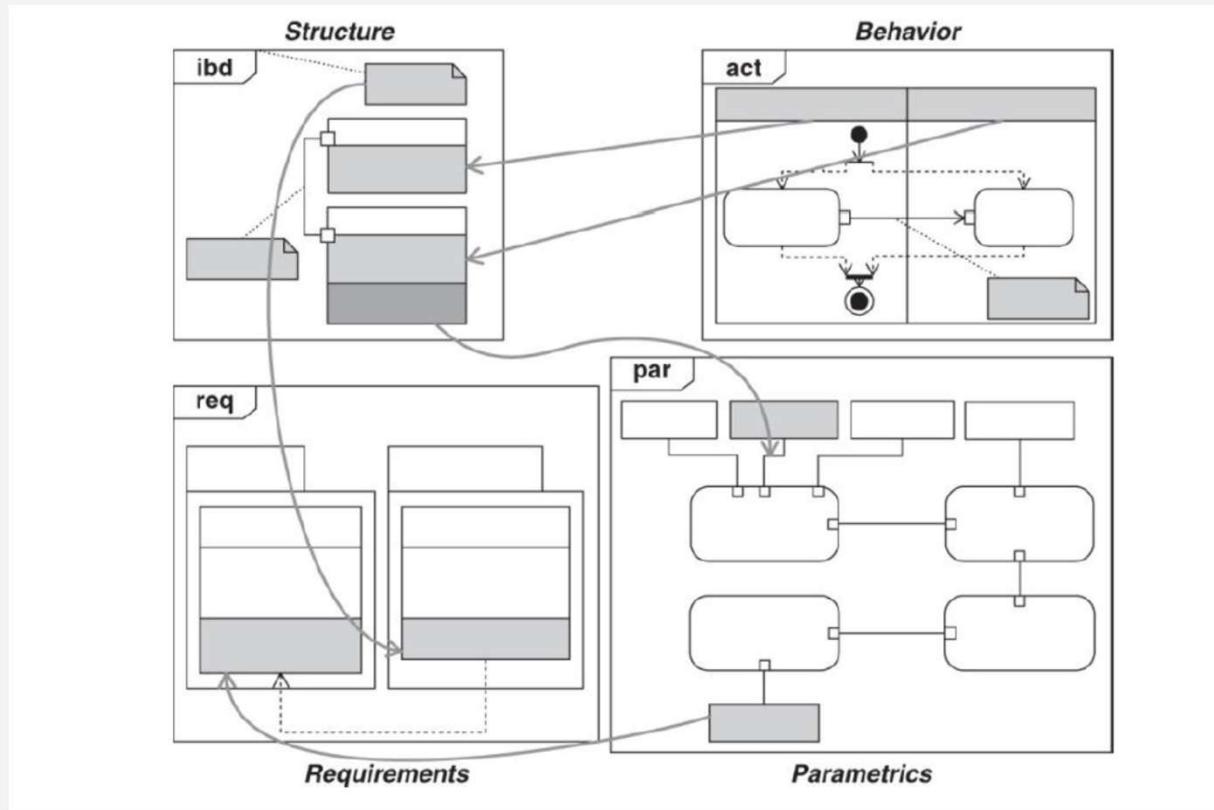
# TOOL FOR SYSTEMS MODELING



**MBSE environment for system architects, system analysts, developers, quality engineers, etc.**

- One of the most popular UML / SysML tools in the market
- Widely regarded as the most standard-compliant tool
- Designed for customization to customer needs

# LES 4 PILIERS DE SYSML



# BIBLIOGRAPHIE

- Khalid Kouiss « Cours IS » Master exécutif UM6P
- Jean-Yves BRON, "System Requirements Engineering - A SysML Supported Requirements Engineering Method", ISTE Press - Elsevier, 2020,
- Caignot, V. Crespel, M. Dérumaux, C. Garreau, P. Kaszynski, B. Martin & S. Roux, « Sciences industrielles de l'ingénieur, MPSI-PCSI-PTSI », ISBN : 978-2-311-01305-4 © Vuibert – août 2013.
- AFIS – « Découvrir et comprendre l'Ingénierie Système » - 2012 – ISBN 9782364930056
- AFIS – Association Française d'Ingénierie Système – [www.afis.fr](http://www.afis.fr)
- INCOSE – International Council on Systems Engineering – [www.incosse.org](http://www.incosse.org)
- JD Piques - SysML for embedded automotive systems: SysCARS methodology, ERTS (Embedded Real Time Software and Systems ) 2014.
- Daniel KROB, « CESAMES Systems Architecting Method », A Pocket Guide, January 2017