

# Réalisation d'une application logicielle.

Jean-Louis BOULANGER



## Plan

- Cout du logiciel
- Processus de réalisation d'une application logicielle
- Ingénierie des exigences

## Cout du logiciel

## Exemple

- Le coût d'une erreur peut dépasser largement le coût du système
    - D'après le cabinet de conseil en technologies de l'information Standish Group International, les défaillances causées par des problèmes de logiciel ont coûté l'an dernier aux entreprises du monde entier environ 175 milliards de dollars, soit **deux fois plus** au moins qu'il y a deux ans.
- Le Monde – 23 oct. 2001

## Répartition des coûts ...

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>◆ Répartition des coûts de développement</li> <li>■ spécification : 6%</li> <li>■ conception : 5%</li> <li>■ codage : 7%</li> <li>■ tests et validation : 15%</li> <li>■ maintenance : 67%</li> </ul> | <ul style="list-style-type: none"> <li>◆ Coûts de correction des erreurs provenant</li> <li>■ exigences et spécification : 56%</li> <li>■ conception : 24%</li> <li>■ codage : 10%</li> <li>■ autres : 10%</li> </ul> |
|--|---|

## Le coût des « bug »

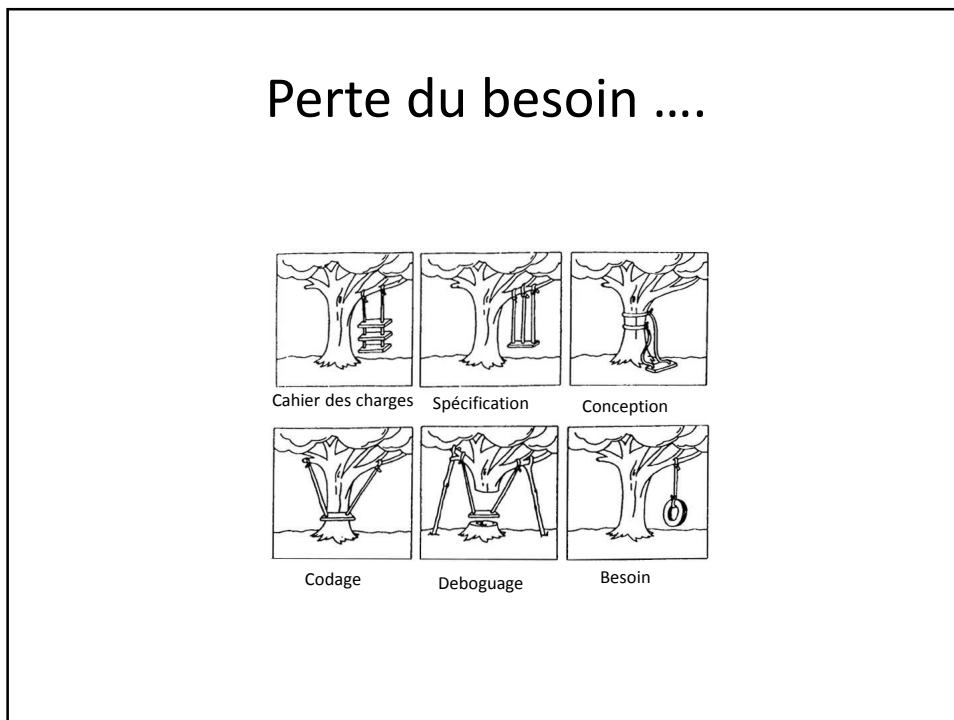
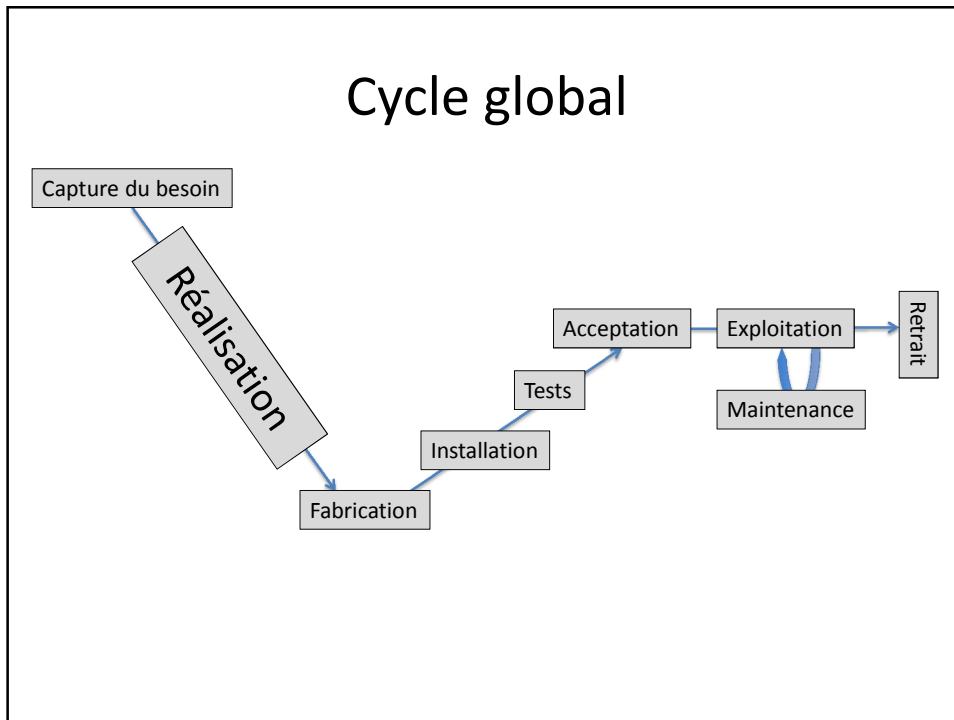
- D'après un rapport du NIST (Assesses Technical Needs of Industry to Improve Software-Testing) daté de 2002 qui porte sur les états unis:
  - Le coût des erreurs et des « bugs » logiciels représente annuellement 59.5 billion de dollars.
- Cette étude indique aussi
  - que tous les « bugs » ne peuvent être corrigés,
  - mais que l'amélioration du processus de vérification permettrait d'en éviter un tiers (soit 22.2 billion de dollars).

LO19 Génie logiciel : Introduction

7

## Processus de réalisation d'une application logicielle

Introduction



# Processus de réalisation d'une application logicielle

CdCF

## MOA/MOE

- Maîtrise d'ouvrage
  - C'est une personne ou une entreprise qui a un besoin et qui consulte des entreprises pour trouver une solution.
- Maîtrise d'œuvre
  - C'est une personne ou plus généralement une société qui va être chargée de réaliser les travaux que le cahier des charges a fixés.
  - Elle va assurer des prestations techniques, qui aboutiront à la mise en œuvre de la solution acceptée par la maîtrise d'ouvrage.

## CdCF (1/2)

- Le Cahier des Charges Fonctionnel (CdCF) d'un projet est un document par lequel la maîtrise d'ouvrage exprime son besoin pour le projet.
- **Définition AFNOR :**
  - Document par lequel le demandeur exprime son besoin (ou celui qu'il est chargé de traduire) en terme de fonctions de services et de contraintes.
- Le CdCF constitue une référence contractuelle entre les parties.

## CdCF (2/2)

Il existe des normes (**AFNOR NF X 50 151**) qui fixe le périmètre du CdCF

- **Identification** Cahier des charges fonctionnel
- **Destinataires.**
  - descriptif du client;
  - descriptif fournisseur.
- **Objet**
  - Définir les fonctionnalités du système à concevoir.
  - Les domaines d'application.
  - Les personnes impliquées dans le projet.
  - Les exigences du client dans la forme et le fond.
  - Base contractuelle entre le client et le fournisseur.

## Exemple de CdCF: la 2CV

« Faites étudier par vos services, une voiture pouvant transporter deux cultivateurs en sabots, cinquante kilos de pomme de terre ou un tonnelet à une vitesse de 60 km/h, pour une consommation de 3 litres au cent. La voiture pourra passer dans les plus mauvais chemins; elle devra pouvoir être conduite par une conductrice débutante et avoir un confort irréprochable. Son prix devra être inférieur au tiers de la traction avant 11 CV. Le point de vue esthétique n'a aucune importance. »

établi par Mr P.J. BOULANGER (responsable de Citroën)  
à Mr BROGLY (Directeur du Bureau d'études), 1936

15

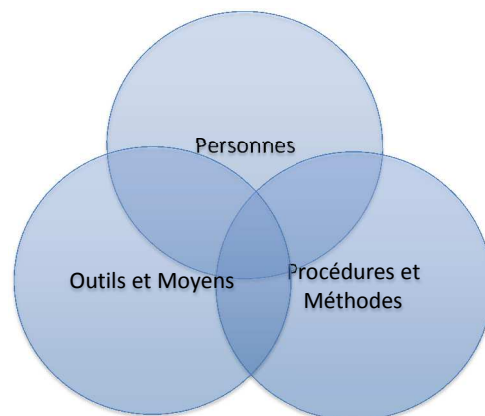




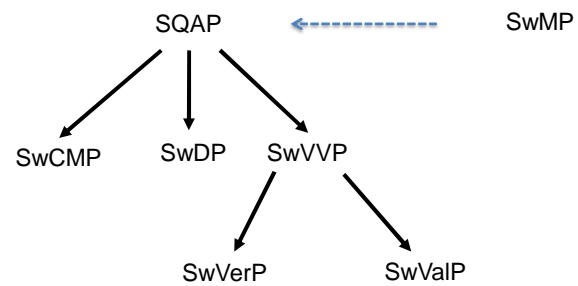
# Processus de réalisation d'une application logicielle

Maitrise de la qualité

## POP



## Processus et méthodes



QAM + Procédures + Templates + Guides

## Processus de réalisation d'une application logicielle

Modèle de maturité

## Définitions

- **Assurance qualité** : Mise en œuvre d'un ensemble approprié de dispositions pré-établies et systématiques destinées à donner confiance en l'obtention d'une qualité requise.
- **Manuel qualité** : Document décrivant les dispositions générales prises par l'entreprise pour obtenir la qualité de ses produits ou de ses services.
- **Plan qualité logiciel** : Document décrivant les dispositions spécifiques prises par une entreprise pour obtenir la qualité du produit ou du service considéré.

Normes ISO et IEEE  
Maîtrise de la Qualité ISO 9001

Evaluation de maturité des processus:

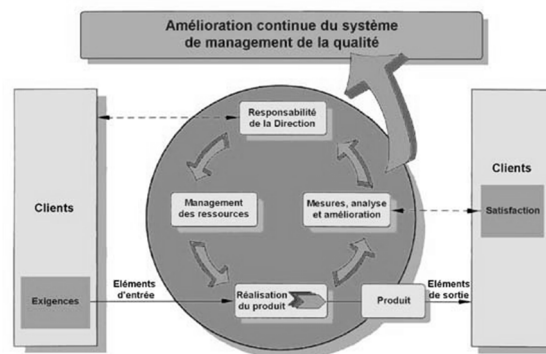
CMM,  
SPICE

- Elles ne prescrivent pas des méthodes précises (ex: OO, CleanRoom, ...);
- Elles proposent un ensemble d'activités requises pour produire des biens de qualité;

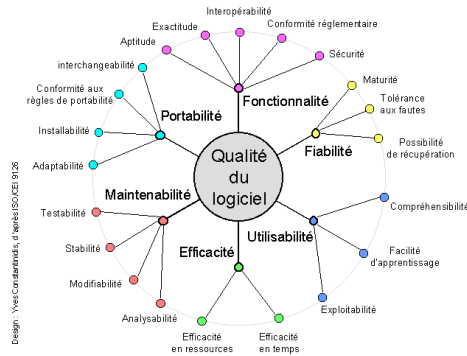
## Principe de base

« Dîtes ce que vous faites,  
faites ce que vous dites,  
et montrez que vous l'avez fait »

## ISO 9001



## Qualité du logiciel IEC 9126



## CMMI 1/5

- CMMi (Capability Maturity Model for integration) est un modèle de maturité dédié à l'industrie du logiciel qui regroupe un ensemble de bonnes pratiques à mettre en œuvre dans les projets de développement.
- Ce modèle est un modèle de référence international d'évaluation et d'amélioration des processus de développement et de maintenance des logiciels. Le déploiement de CMMi au sein d'une entreprise s'inscrit dans une démarche d'amélioration continue et doit par conséquent être adapté aux différentes organisations et typologies de projets.
- Voir le site du Software Engineering Institute : <http://www.sei.cmu.edu/cmmi>

## CMMI 2/5

- CMMi apporte une réponse cohérente dans le cadre de la relation MOA-MOE.
- Il permet :
  - la satisfaction des utilisateurs finaux au travers de la gestion des exigences explicites et implicites ;
  - la maîtrise des projets en termes de délais, de budget et de qualité ;
  - la transparence et une vision claire et partagée des projets par toutes les parties prenantes.

## CMMi 3/5

CMMi regroupe des pratiques par niveau de maturité, sur une échelle allant de 1 (le plus basique) à 5 (le plus élaboré). A chaque niveau de maturité correspond des domaines de processus.

Les cinq niveaux de maturité et domaines de processus de CMMi sont :

- **Niveau 5 - Optimisé**
  - Innovation et déploiement organisationnels
  - Analyse causale et résolution
- **Niveau 4 - Maîtrisé**
  - Performance du processus organisationnel
  - Gestion quantitative du projet
- **Niveau 3 - Défini**
- **Développement des exigences**
  - Solution technique
  - Intégration du produit
  - Vérification
  - Validation
  - Focalisation sur le processus organisationnel
  - Définition du processus organisationnel
  - Formation organisationnelle
  - Gestion de projet intégrée
  - Gestion des risques
  - Analyse et prise de décision
- **Niveau 2 - Reproductible**
  - Gestion des exigences
  - Planification de projet
  - Suivi et contrôle projet
  - Gestion des accords avec les fournisseurs
  - Mesure et analyse
  - Gestion de configuration
  - Assurance qualité du processus et du produit
- **Niveau 1 - Initial**

- Pour les exigences, CMMi décrit des bonnes pratiques pour le **développement des exigences** (au niveau 3) et la **gestion des exigences** (au niveau 2). CMMi introduit les notions d'exigences client et d'exigences produit, et présente des activités à la fois spécifiques et communes à ces deux niveaux d'exigences.

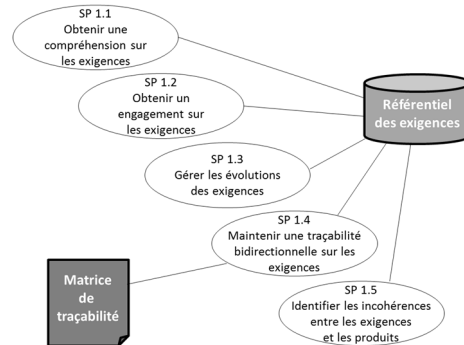
## « Développement des exigences »

Le domaine de processus *Requirements Development* est décomposé en 3 sous-domaines.

- **SG 1 Développer les exigences client**
  - Capturer les besoins
  - Développer les exigences client
- **SG 2 Développer les exigences produit**
  - Etablir les exigences produit et les exigences des composants
  - Allouer les exigences des composants du produit
  - Identifier les exigences d'interfaces
- **SG 3 Analyser et valider les exigences**
  - Etablir les concepts et les scénarios opérationnels
  - Etablir une définition des fonctionnalités requises
  - Analyser les exigences
  - Evaluer les risques sur les exigences
  - Valider les exigences

# Gestion des exigences

- **SG 1 Gérer les exigences**
- Obtenir une compréhension sur les exigences
- Obtenir un engagement sur les exigences
- Gérer les évolutions des exigences
- Maintenir la traçabilité bidirectionnelle sur les exigences
- Identifier les incohérences entre les produits et les exigences



# Unified process - UP

- **Bonne pratique 1 : Gestion des exigences avec des cas d'utilisation (Use Cases)**

Le processus de développement est centré sur l'utilisateur. Les cas d'utilisation vont représenter les fonctions essentielles du futur système. Ils sont utilisés pour le recueil des exigences initiales (fonctionnelles et techniques) et pour la définition du périmètre de l'application. Comme nous l'avons déjà vu au chapitre 7 sur la modélisation basée sur les cas d'utilisation, nous pourrions organiser hiérarchiquement les cas d'utilisation et mettre en facteur des cas d'utilisation.

- **Bonne pratique 2 : Développement itératif piloté par les risques**

Le projet global est découpé en « mini-projets » appelés « itérations » et qui donnent lieu à un « incrément ». Chaque itération comprend un certain nombre de cas d'utilisation et doit traiter en priorité les risques majeurs. Une itération reprend les livrables dans l'état où les a laissés l'itération précédente et les enrichit progressivement (de façon incrémentale). Les itérations sont regroupées dans une phase et chaque phase est ponctuée par un jalon qui marquera la décision que les objectifs fixés préalablement ont été atteints.

- **Bonne pratique 3 : Architecture à base de composants**

L'architecture regroupe les différentes vues du système qui doit être construit et prévoit la réalisation de tous les cas d'utilisation. L'architecture et les cas d'utilisation évoluent de façon concomitante. Les premières itérations définissent le noyau de l'architecture.

- **Bonne pratique 4 : Modélisation visuelle avec UML**

A partir des cas d'utilisation, les concepteurs et développeurs créent une série de modèles UML. Ces modèles permettent de prendre en compte des situations exceptionnelles et de préparer la phase d'analyse.

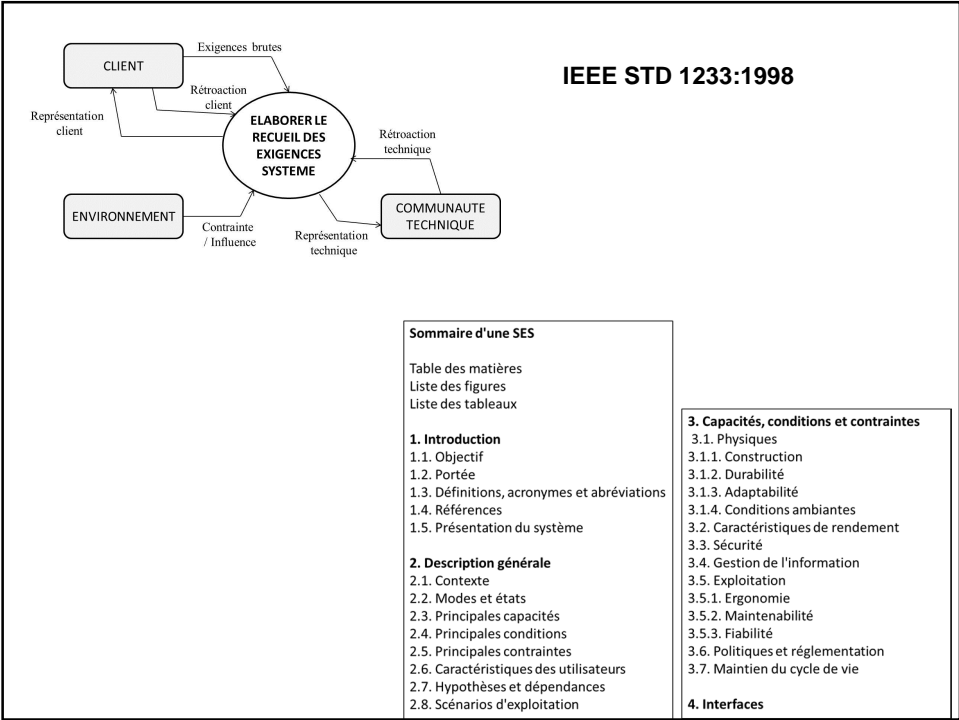
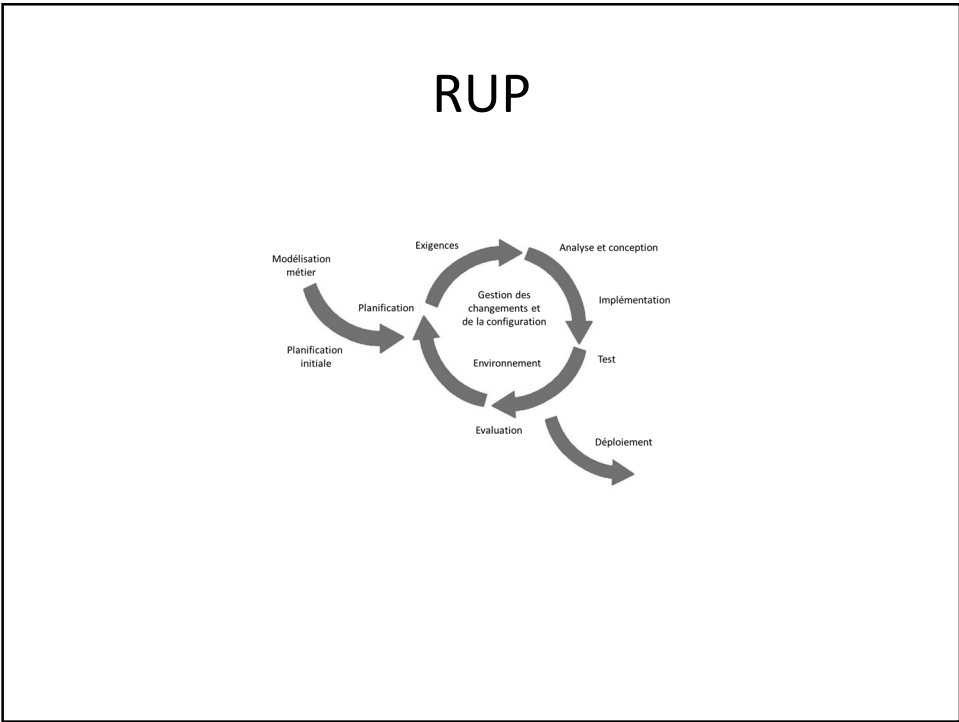
- **Bonne pratique 5 : Vérification permanente de la qualité**

Le test d'une partie importante du projet à tout moment, et au fur et à mesure de son avancement, va assurer une livraison d'un produit de qualité.

- **Bonne pratique 6 : Gestion des changements**

Les projets sont souvent créés par des équipes nombreuses, parfois distantes, et qui utilisent différentes plates-formes de développement. Par conséquent, il est essentiel de s'assurer que les modifications apportées à un système sont synchronisées et contrôlées en permanence.





# IEEE std 830:1998

- Les buts d'une Spécification d'Exigences du Logiciel (SEL) sont :
  - Établir un accord entre client et fournisseur sur ce qu'il faut faire ;
  - Réduire les coûts de développement du logiciel ;
  - Fournir une base pour l'évaluation des coûts et de l'échéancier ;
  - Fournir une base pour la vérification et la validation (V&V) ;
  - Faciliter les transferts vers d'autres services, d'autres personnes, etc.
  - Fournir une base pour les améliorations.

## Sommaire d'une SEL

### 1 INTRODUCTION

- 1.1 Objet du document
- 1.2 Vision et portée du produit
- 1.3 Définitions, acronymes et abrég.
- 1.4 Références
- 1.5 Organisation du document

### 2 DESCRIPTION GÉNÉRALE

- 2.1 Contexte ("perspective du produit")
- 2.2 Synthèse des fonctions
- 2.3 Caractéristiques des utilisateurs
- 2.4 Contraintes
- 2.5 Hypothèses et dépendances

### 3 EXIGENCES DÉTAILLÉES

- 3.1 Exigences pour les interfaces externes
  - 3.1.1 Interfaces utilisateur
  - 3.1.2 Interfaces matérielles
  - 3.1.3 Interfaces logicielles
  - 3.1.4 Interfaces de communication
- 3.2 Exigences fonctionnelles
- 3.3 Exigences de performance
- 3.4 Contraintes de conception
- 3.5 Exigences qualité (ISO 9126)
- 3.6 Autres exigences

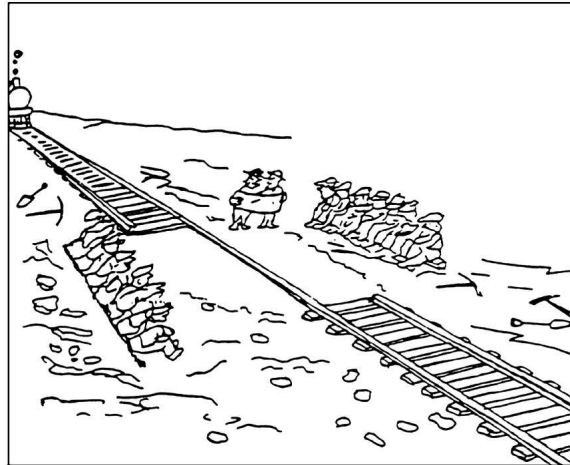
#### Appendices

- A. Glossaire
- B. Modèles d'analyse
- C. Priorité des exigences
- D. Matrice de traçabilité

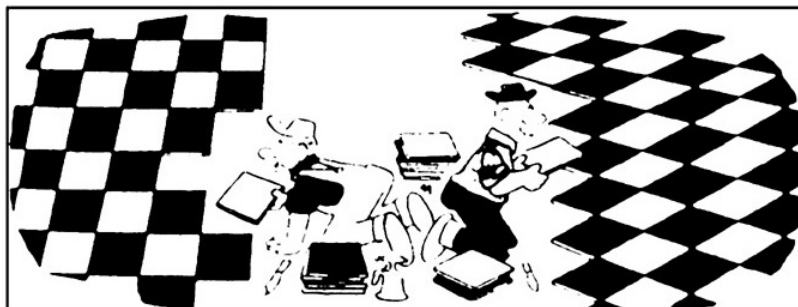
# Processus de réalisation d'une application logicielle

Cycle de réalisation

## Mauvaise gestion



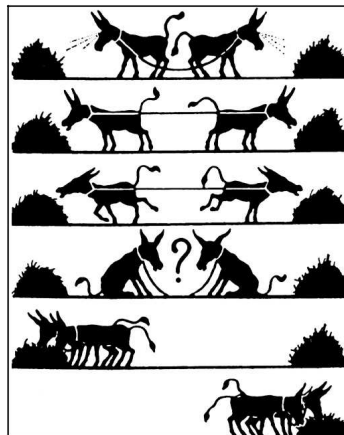
## Mauvaise spécification



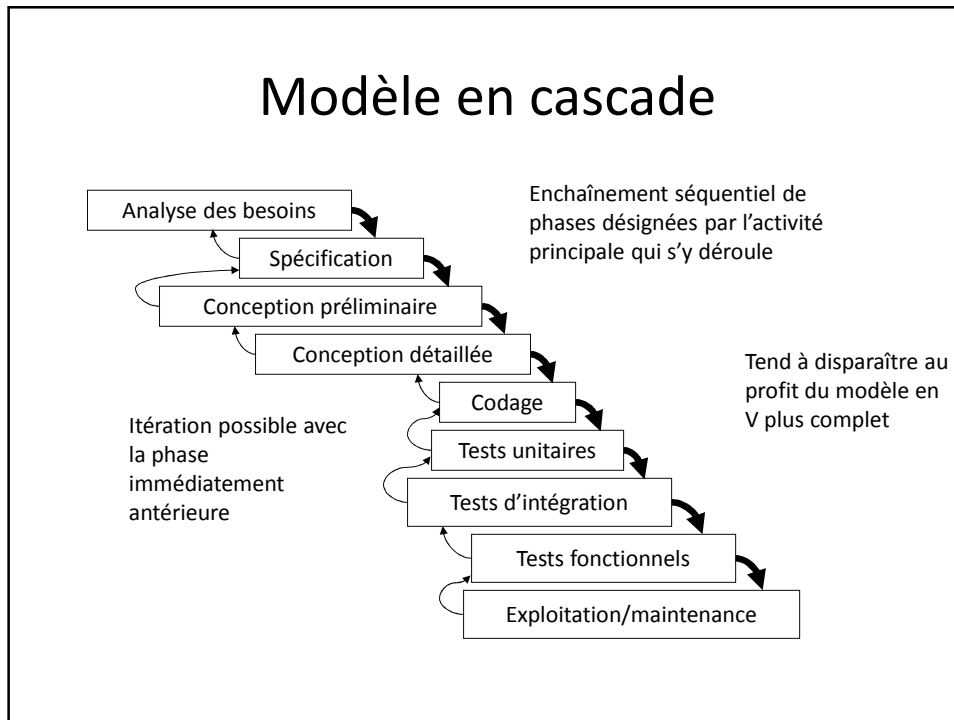
## Aspects organisationnels

- On ne parvient à rien en laissant libre cours à la créativité débridée de chacun.
- Organiser est également nécessaire pour parvenir à une unité de conception.
- Le management fait donc également partie du processus de réalisation.

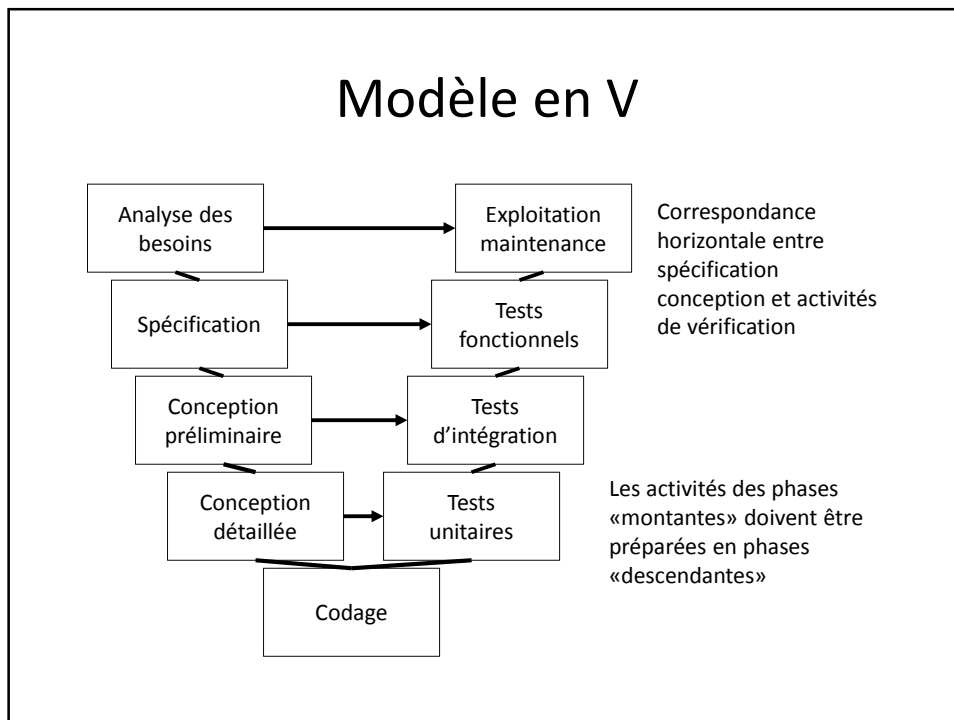
## Vertu de l'organisation



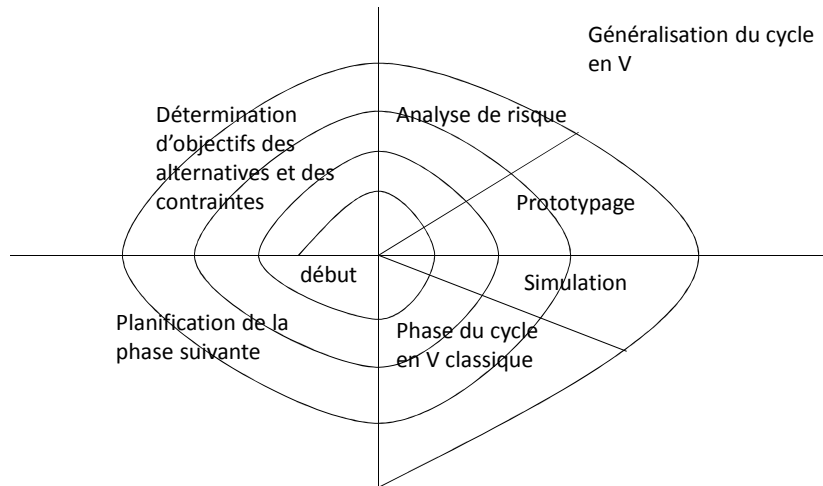
## Modèle en cascade



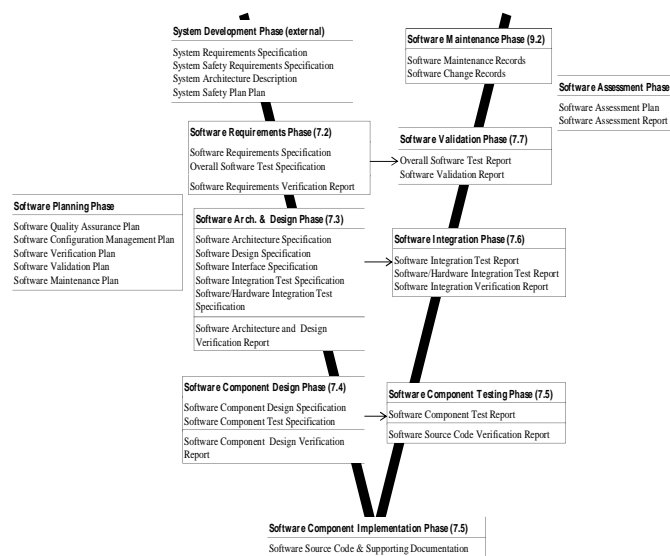
## Modèle en V



# Modèle en spirale (Boehm 1988)



# CENELEC 50128:2011



# Ingénierie des exigences

# Ingénierie des exigences

Introduction

## Rappel

- Plus de 2/3 du coût final d'un système est déterminé au moment de la formalisation des exigences.
  - Un défaut détecté dès la spécification coûte 40 fois moins cher à corriger que s'il est détecté en phase de validation.
  - La gestion des exigences est à l'origine de 40% des facteurs de réussite d'un projet, d'après le rapport "Chaos" du Standish Group (1994).
  - Une mauvaise ou une absence de gestion des exigences est à l'origine de plus de 30% des facteurs d'échec d'un projet, toujours d'après le rapport "Chaos" du Standish Group (1994).
- [http://www1.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www1.standishgroup.com/sample_research/chaos_1994_1.php)

## Répartition des causes d'échec

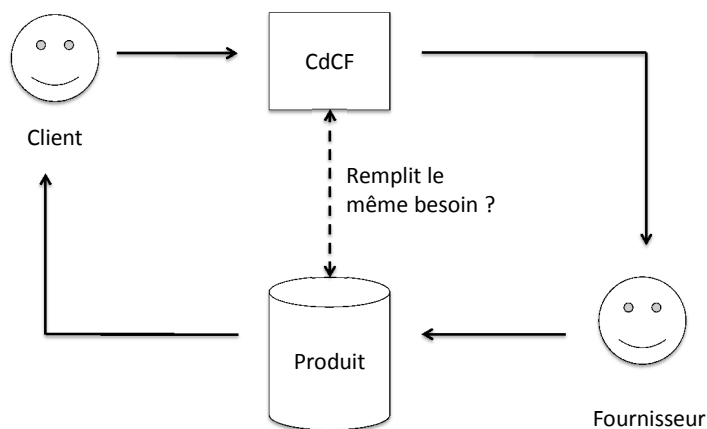
Description	%
Incomplete Requirement	13,1%
Lack of user involvement	12,4%
Lack of Resources	10,6%
Unrealistic expectations	9,9%
Lack of executive support	9,3%
Changing Requirement/specification	8,7%
Lack of planning	8,1%
Didn't need it any longer	7,5%

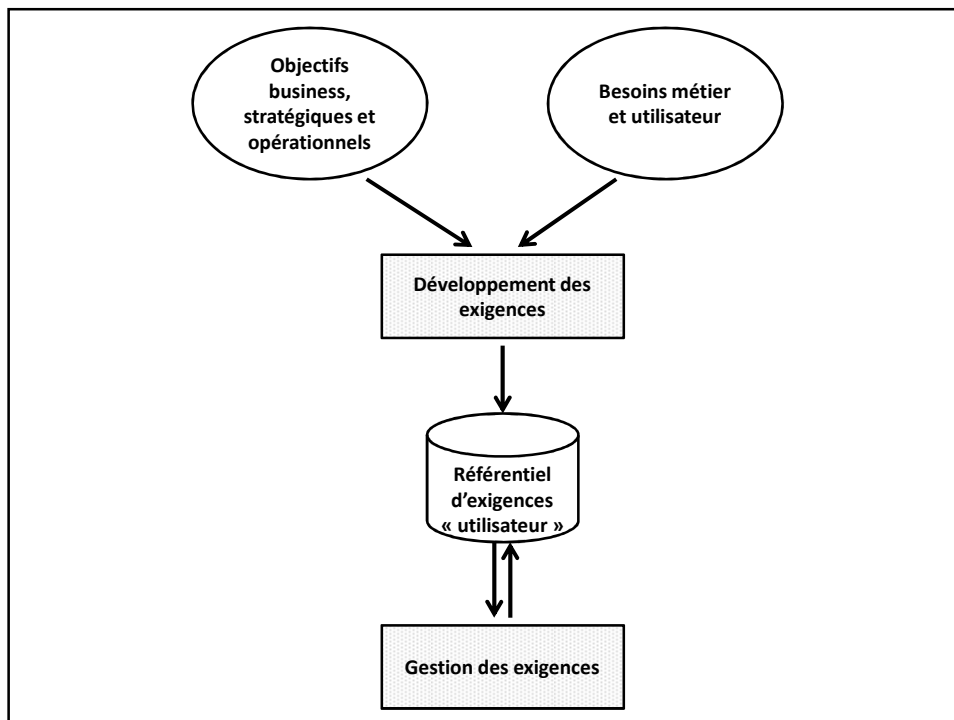
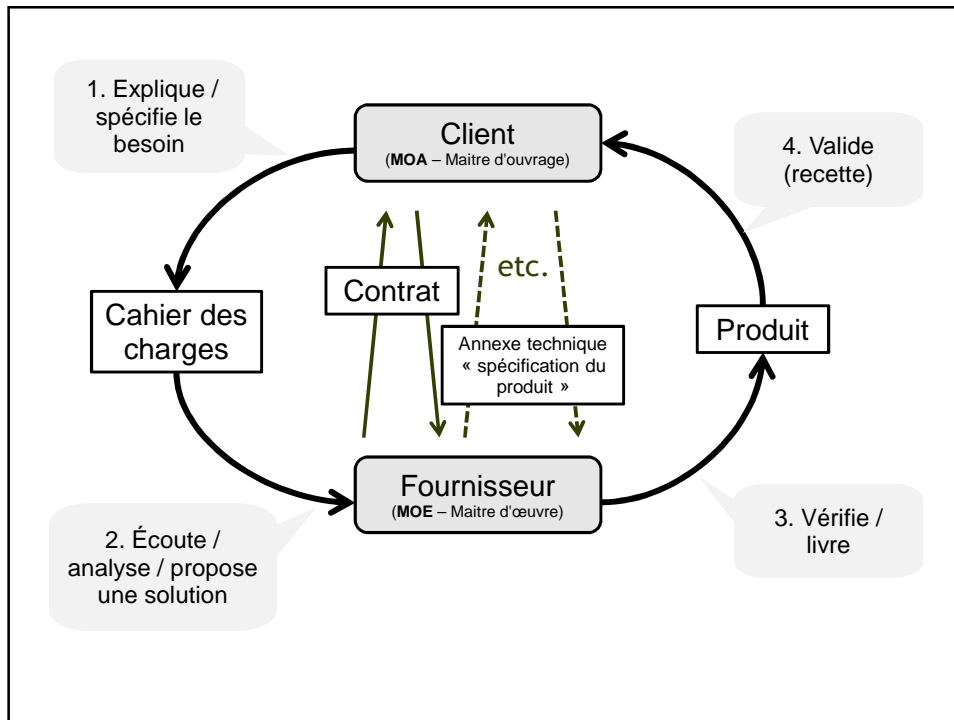


# Ingénierie des exigences

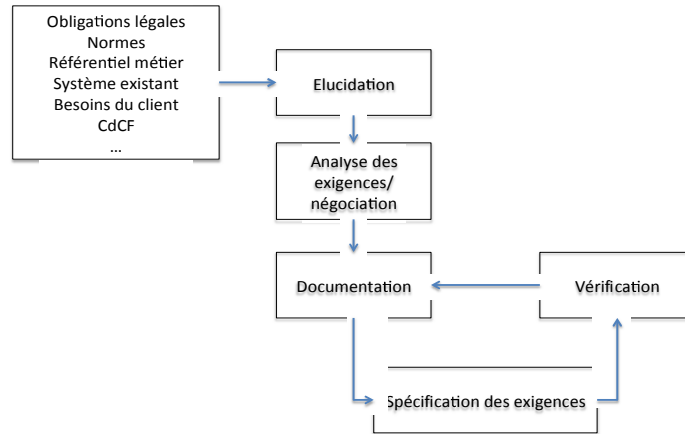
## Méthodologie

### Client vs Fournisseur

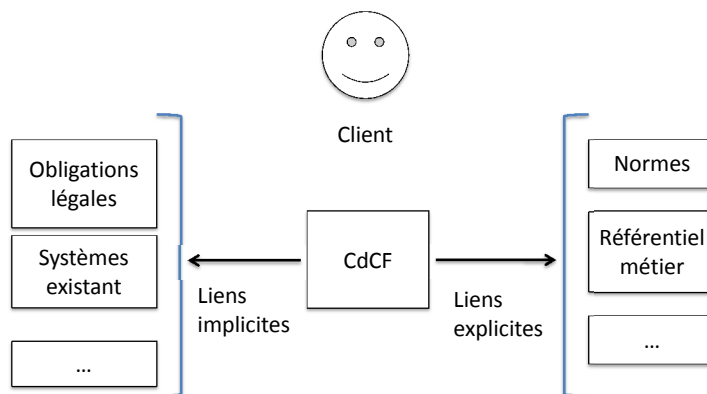




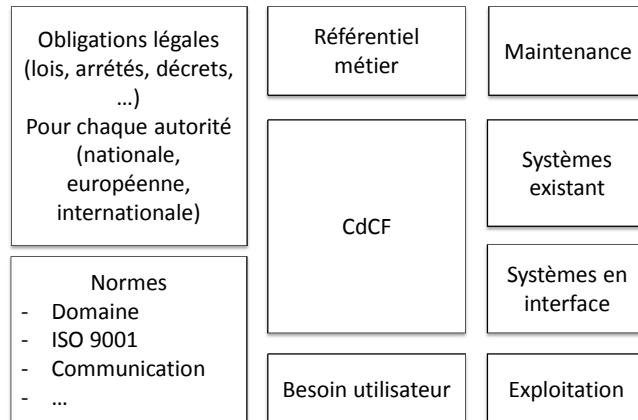
## Processus global



## Sources 1/2



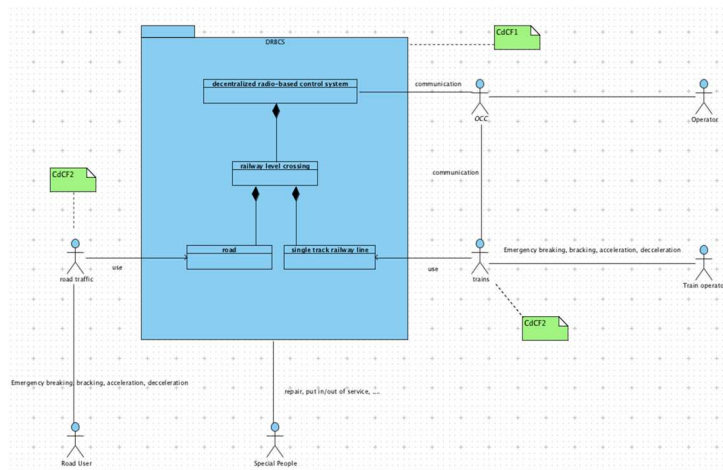
## Sources 2/2



## Défauts typiques d'un CDCF

- **Bruit:** texte n'apportant pas de nouvelle information
- **Silence:** caractéristique non précisée
- **Sur-Spécification:** solution proposée
- **Contradiction:** incompatibilité entre deux caractéristiques
- **Ambiguïté:** plusieurs interprétations possibles
- **Vœu pieu:** caractéristique impossible à réaliser
- **Sous-Entendu:** connaissance présumée du domaine

## Parties prenantes 1/3



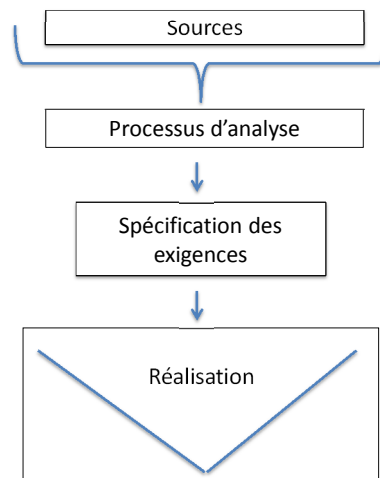
## Parties prenantes 2/3

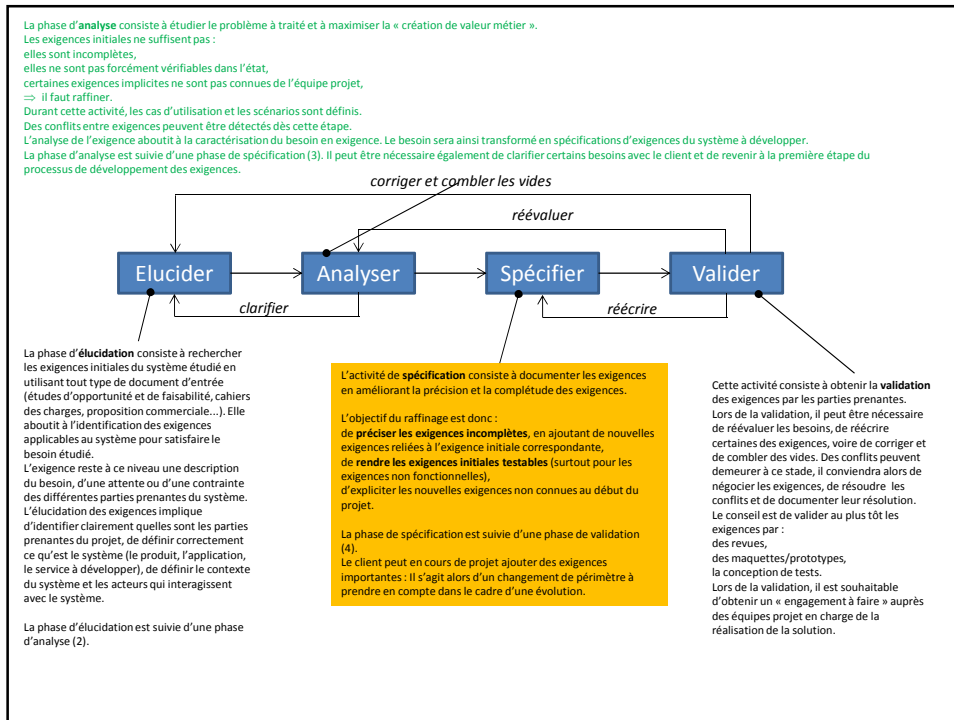
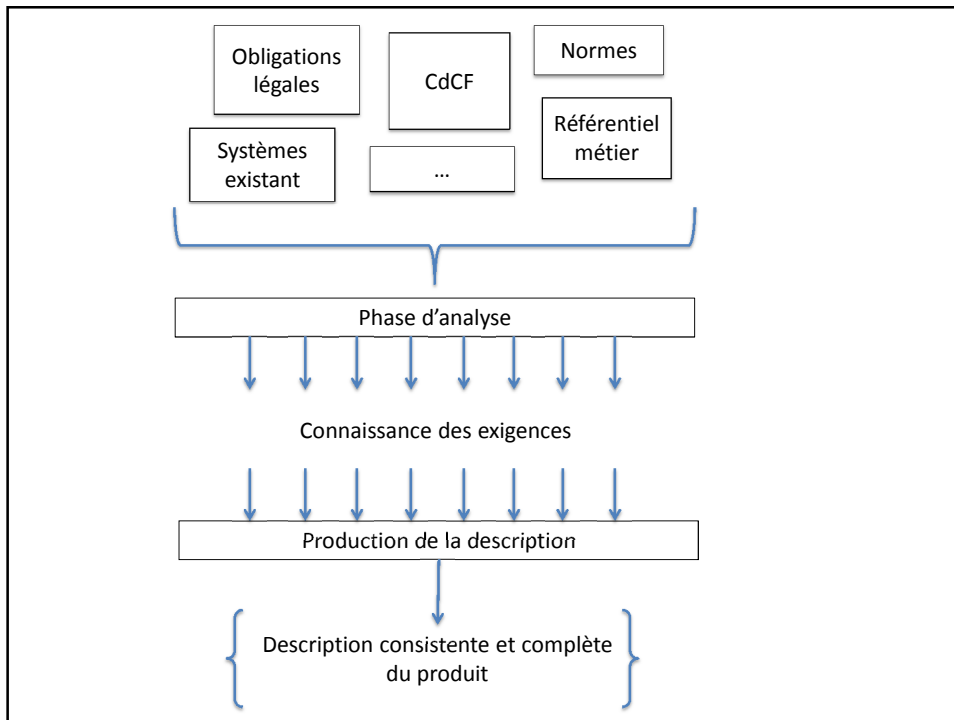
- Une partie prenante est, par définition, une personne ou une entité qui a un intérêt dans le projet. Une manière de faire un tour d'horizon complet des parties prenantes est de se poser les questions suivantes :
  - Qui financent le projet ? Qui disposent du budget ?
  - Y'a-t-il des sponsors pour ce projet ? Si oui, lesquels ?
  - Quels sont les utilisateurs du futur système ?
  - Par utilisateur, on entend bien évidemment l'utilisateur final, mais pas uniquement. Il faut penser à l'exploitation du système (opérateur, administrateur, etc.), à sa maintenance et à son retrait.
  - La figure précédente fait apparaître plusieurs utilisateurs (opérateur, les conducteurs routiers, les conducteurs de trains et la maintenance).
  - Qui sont les personnes qui vont concevoir et tester le système ?
  - Qui va déployer le système et former les utilisateurs finaux ?
  - Y'a-t-il des autorités en charges de l'autorisation du système ?
  - Il peut y avoir des autorités nationale, européenne et internationale avec pour chacune une législation et des organismes représentatifs.
  - Y'a-t-il des systèmes existants en interface ?
  - Y'a-t-il des besoins de certification ?
  - Etc.

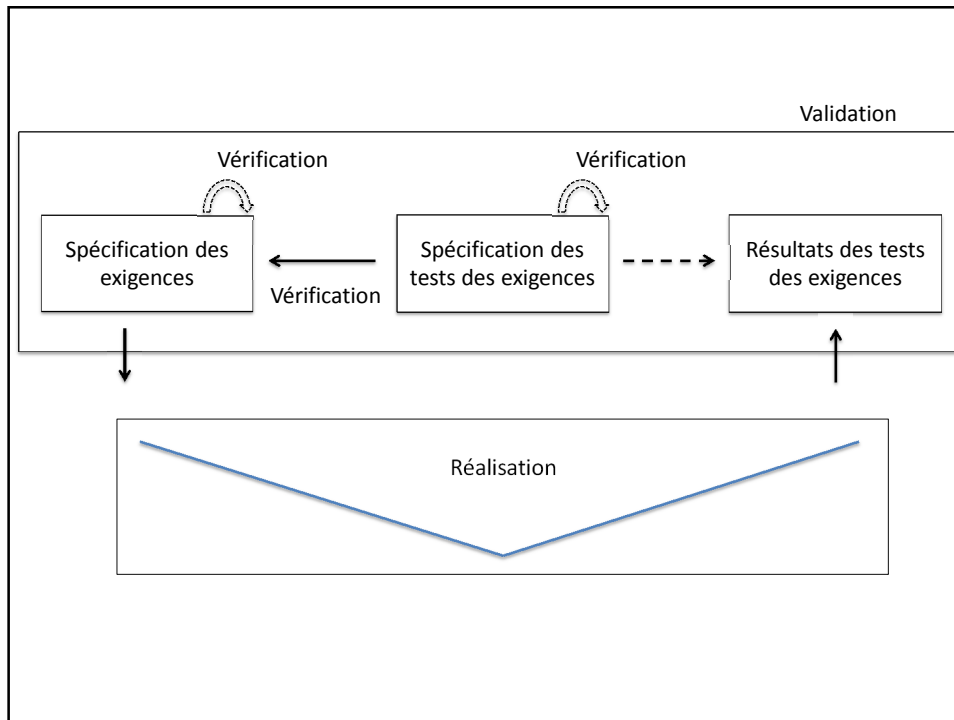
## Parties prenantes 3/3

Nom	Organisme	Rôles	Coordonnées	Disponibilité	Domaine	Niveau d'expertise	Objectif Intérêt dans le projet

## Processus global

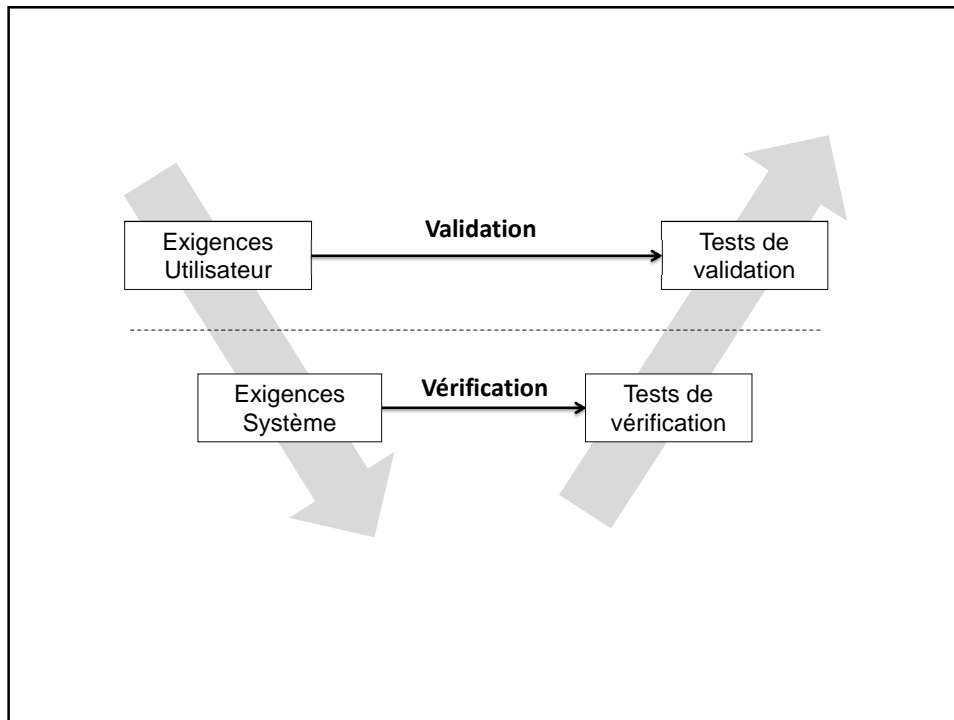






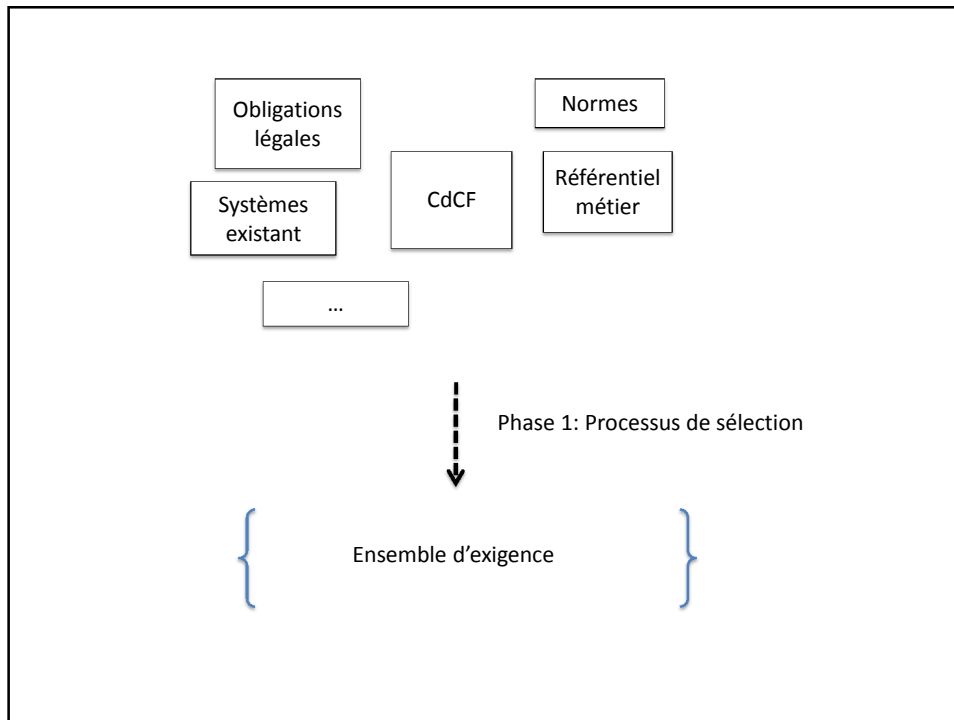
Type de conflit	Description	Exemple
conflit sur le sujet	une même exigence est interprétée différemment par les parties prenantes	ambiguïté d'une exigence de performance
conflit d'intérêts	les objectifs et les buts des parties prenantes divergent	respect des coûts versus haute qualité requise
conflit de valeurs	les parties prenantes valorisent différemment un même critère, divergence de « l'idéal personnel »	codeur/décodeur MP3 et OGG pour un lecteur radio – ou bien MP3 seulement
conflit de structure	le niveau d'influence des parties prenantes en conflit est inégal	un employé / sa hiérarchie





## Ingénierie des exigences

Gestion des exigences



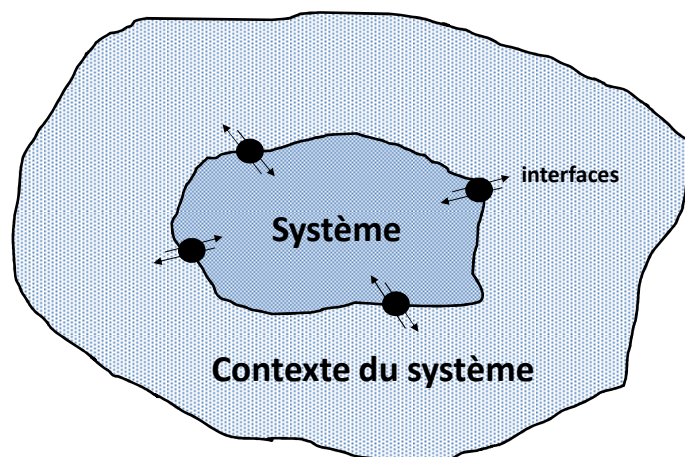
## Identification

- Processus d'élaboration:
  - Expression « d'intuition » (bonnes où mauvaises);
  - Expression d'une connaissance basée sur l'expérience;
  - Expression de propriétés issues d'un travail d'analyse voir de brainstorming;
  - ...
  - Mise en place d'une activité de déduction (semi-)formelle:
    - Basée sur une étude de risque par exemple, ...

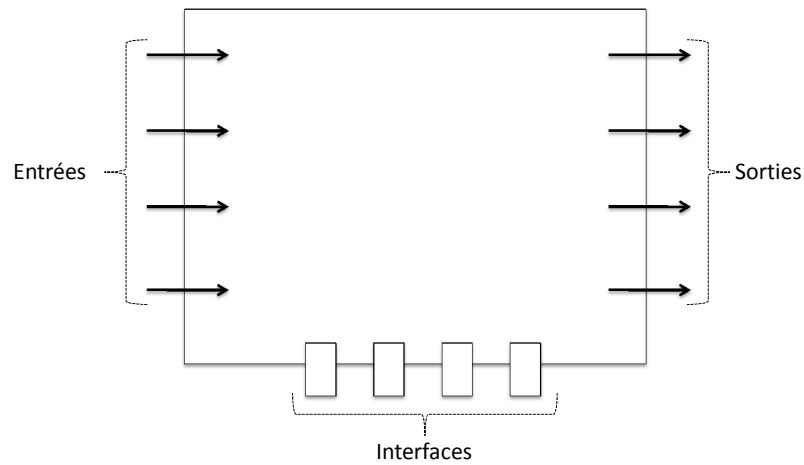
# Ingénierie des exigences

Analyse

## Système et contexte du système



## Identifier la bordure 1/2



## Identification de la bordure 2/2

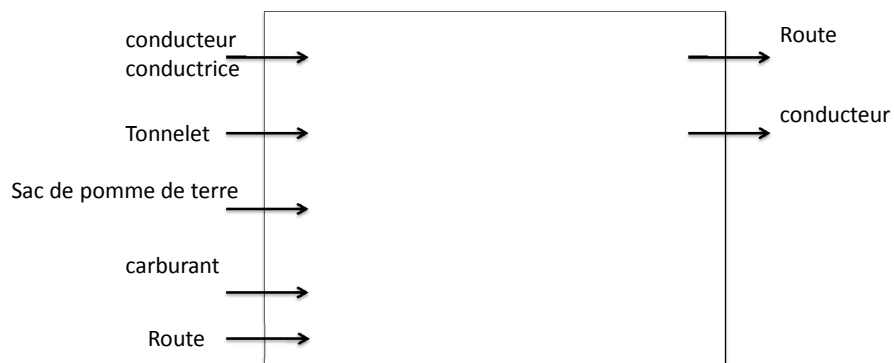
- Personnes en interaction
- Autre systèmes/sous-système/équipement/carte/logiciel en interaction
- ...

## Exemple de CdCF: la 2CV

« Faites étudier par vos services, une voiture pouvant transporter deux cultivateurs en sabots, cinquante kilos de pomme de terre ou un tonnelet à une vitesse de 60 km/h, pour une consommation de 3 litres au cent. La voiture pourra passer dans les plus mauvais chemins; elle devra pouvoir être conduite par une conductrice débutante et avoir un confort irréprochable. Son prix devra être inférieur au tiers de la traction avant 11 CV. Le point de vue esthétique n'a aucune importance. »

73

## 2CV - interfaces



## Éléments à analyser

- Au sein du système, il faut donc identifier :
  - Les interfaces avec l'environnement , c'est interfaces peuvent être
    - électrique,
    - mécanique,
    - pneumatique,
    - logiciel,
    - etc.
  - Les états :
    - à l'arrêt,
    - en marche,
    - état dégradé,
    - Etc.

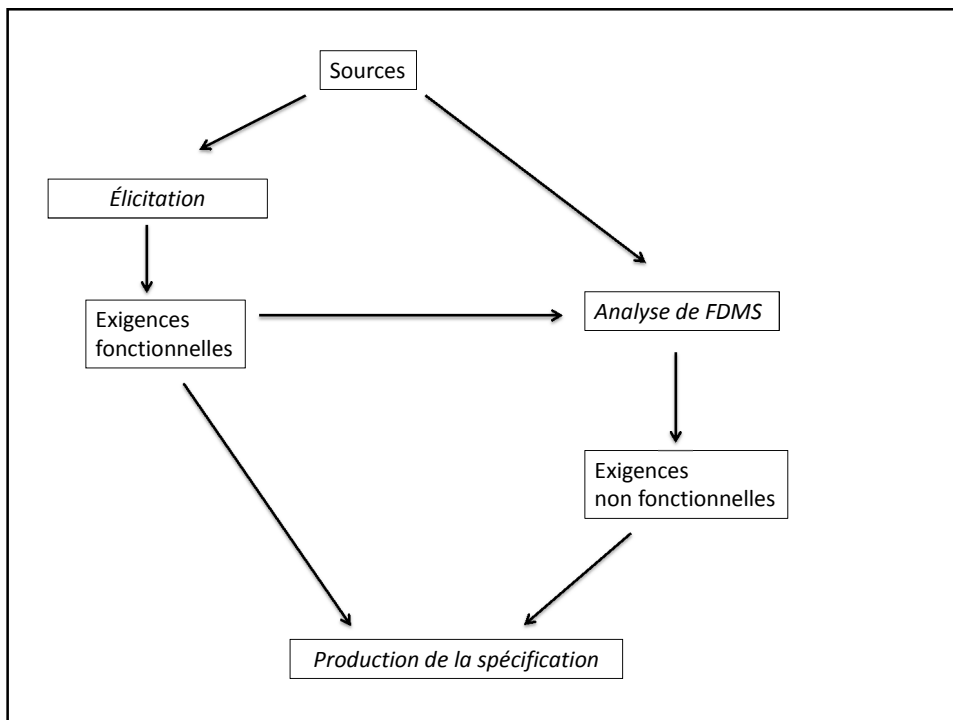
La notion d'état introduit une partition entre les états de bon fonctionnement, les états de repli et les états dangereux ;

- La notion de comportement correct, de comportement dégradé et de comportement dangereux ;
- La notion de besoin fonctionnel et non-fonctionnel.

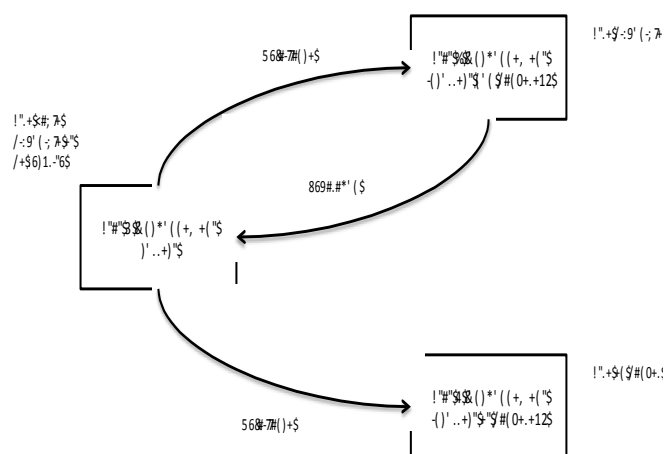
## État du système

La notion d'état introduit une partition entre les états de bon fonctionnement, les états de repli et les états dangereux ;

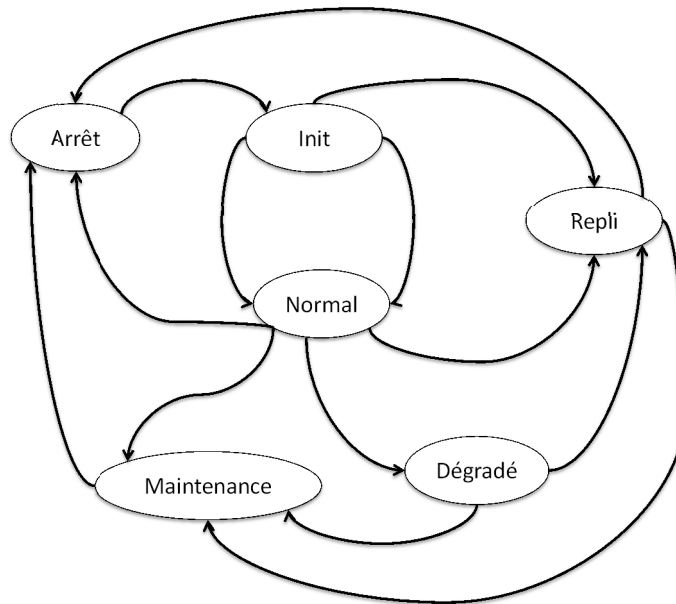
- La notion de comportement correct, de comportement dégradé et de comportement dangereux ;
- La notion de besoin fonctionnel et non-fonctionnel.



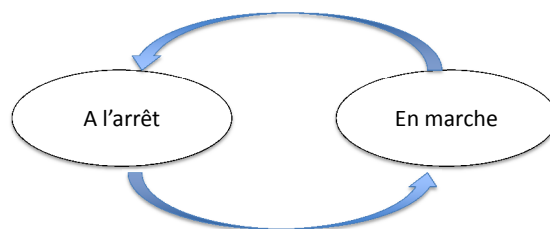
## Différents modes de fonctionnement 1/2



## Différents modes de fonctionnement 2/2

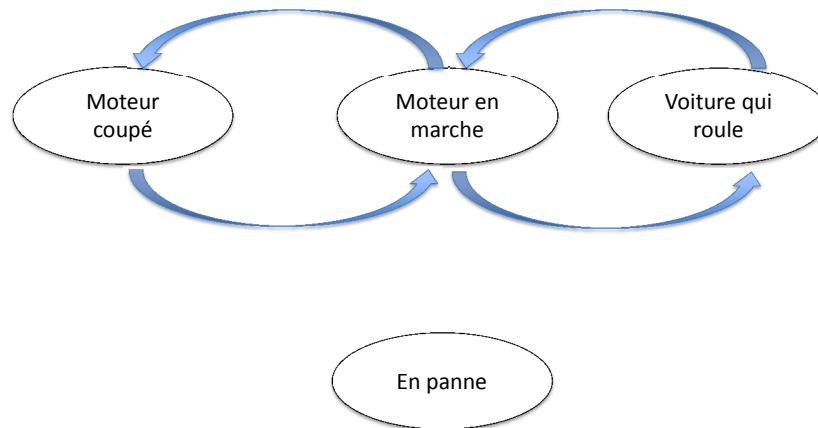


## 2CV – état – première idée





## 2CV – état – avec panne



### Identification des fonctions principales

- Lorsqu'on a identifié la bordure et les éléments en interaction, il est intéressant de tenter d'identifier les fonctions principales du système
- Première possibilité : analyser le CdCF
- Second possibilité : analyser les interactions

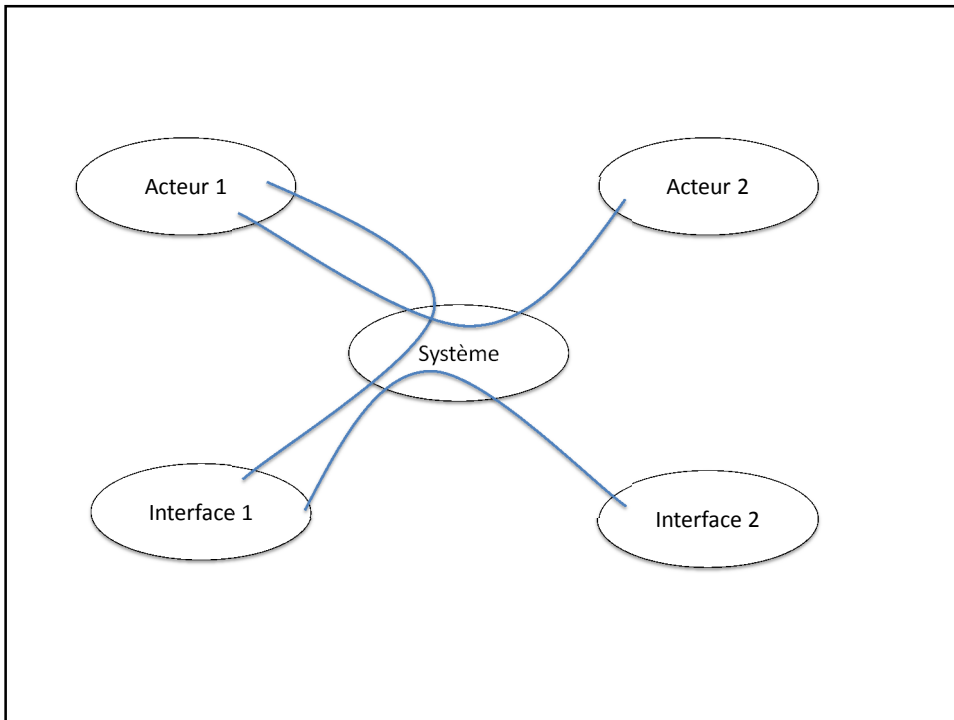
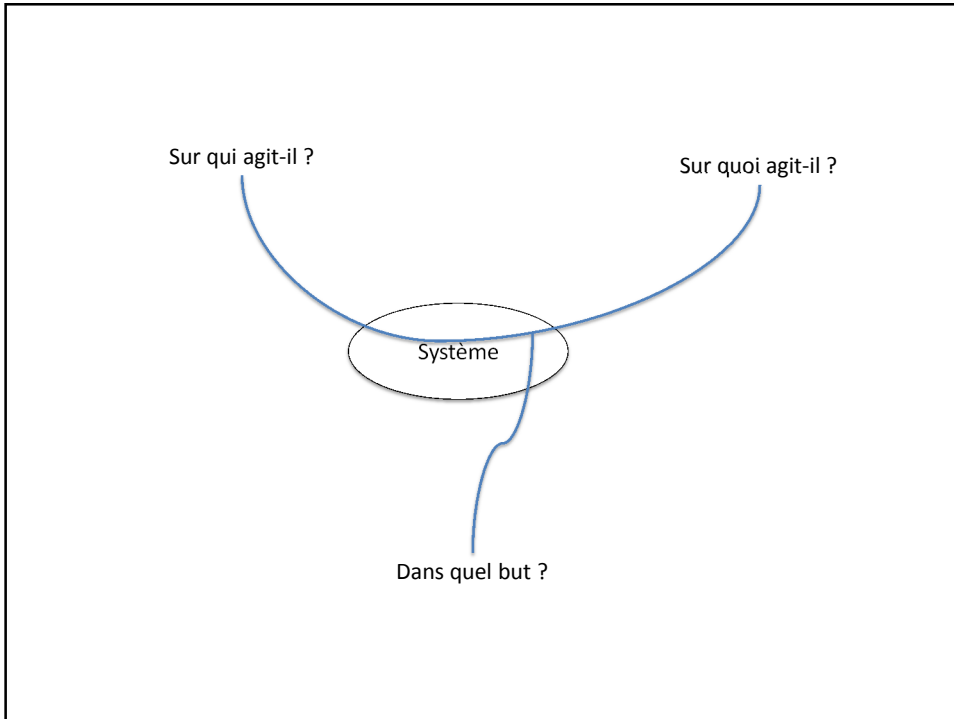
## Exemple de CdCF: la 2CV

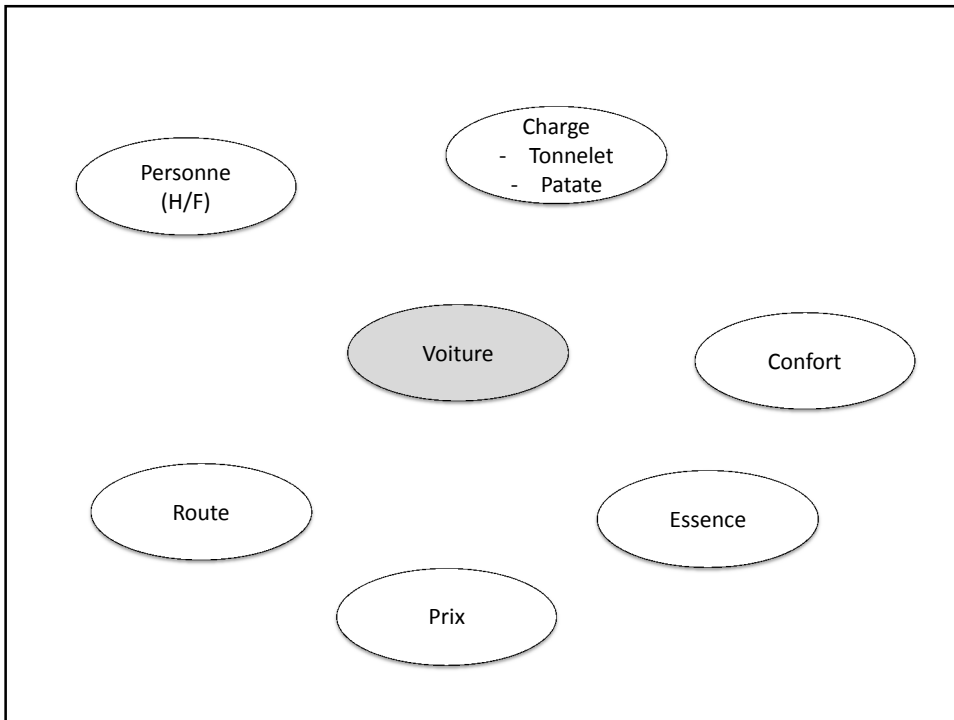
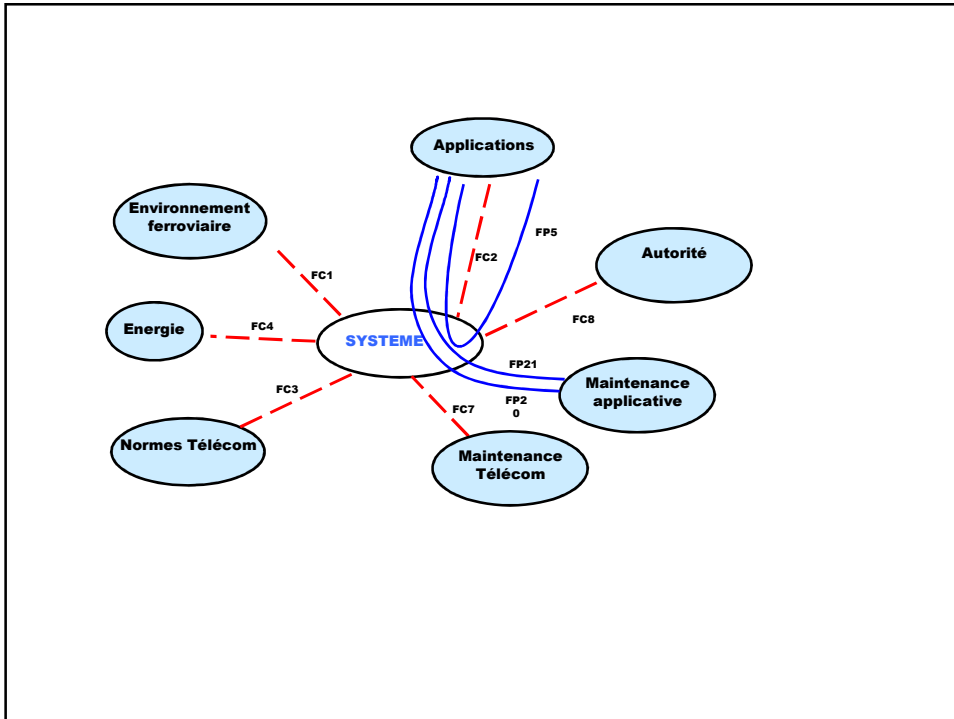
« Faites étudier par vos services, une voiture pouvant transporter deux cultivateurs en sabots, cinquante kilos de pomme de terre ou un tonnelet à une vitesse de 60 km/h, pour une consommation de 3 litres au cent. La voiture pourra passer dans les plus mauvais chemins; elle devra pouvoir être conduite par une conductrice débutante et avoir un confort irréprochable. Son prix devra être inférieur au tiers de la traction avant 11 CV. Le point de vue esthétique n'a aucune importance. »

83

## Après analyse du texte

- Transporter deux cultivateurs en sabots;
- Transporter cinquante kilos de pomme de terre *ou* un tonnelet;
- Consommer 3litre;
- Rouler a 60 Km/H
- Être conduite par une femme débutante
- Avoir un confort

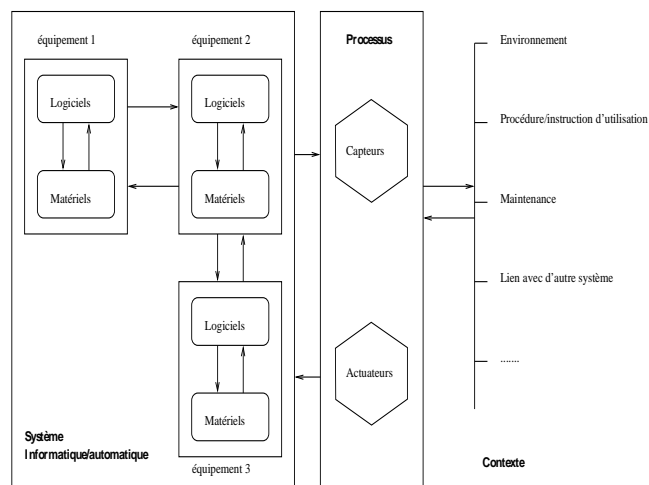


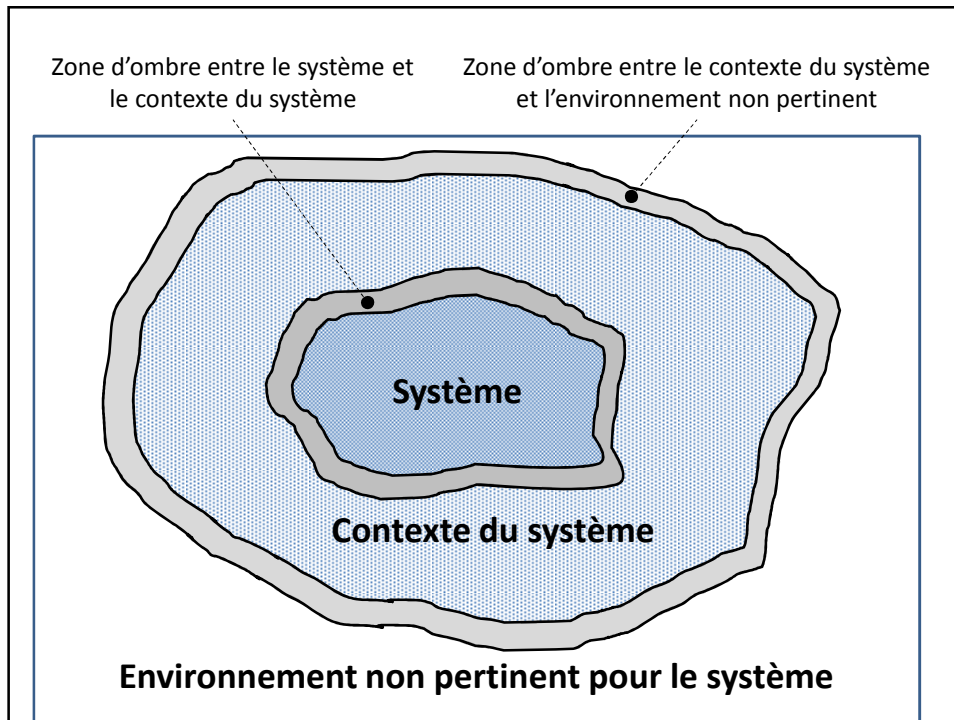


# Ingénierie des exigences

Systeme

## Exemple de système complexe





## Ingénierie des exigences

Bases

## Définition

- **Définition : Exigence**

*Une exigence est un énoncé qui traduit un besoin et/ou des contraintes (techniques, coûts, délais, etc.). Cet énoncé est rédigé dans un langage qui peut être : naturel, mathématique, etc.*

En anglais : **Requirement**

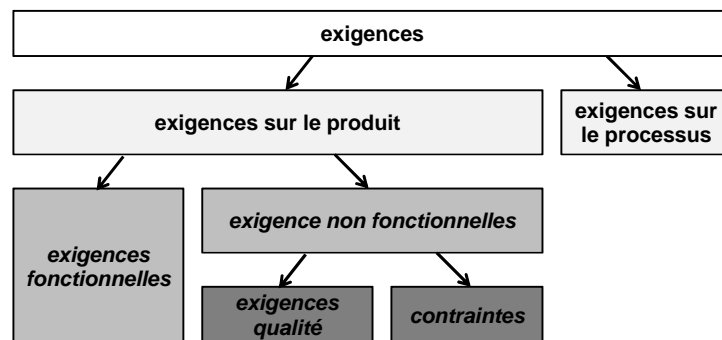
Attributs	Description
ID	Identificateur unique
TEXTE	Texte de l'exigence
SOURCE	Élément ayant permis d'introduire cette exigence

## Ingénierie des Exigences

- Les besoins et les exigences devraient exprimer  
le **Quoi** et non le **Comment**
- Les clients ont des besoins, ces besoins sont traduits en exigences qui guident la spécification  
... ces besoins peuvent s'exprimer à chaque étape du cycle de vie
- Tous les besoins exprimés par le client doivent être implantés (ou pris en compte)  
→ il faut tracer cette prise en compte

## Complément

- Selon l'AFIS: Une **exigence** prescrit une propriété dont l'obtention est jugée nécessaire. Son énoncé peut être une fonction, une aptitude, une caractéristique ou une limitation à laquelle doit satisfaire un système, un produit ou un processus.





## 2CV – Fonctionnel - Non fonctionnel

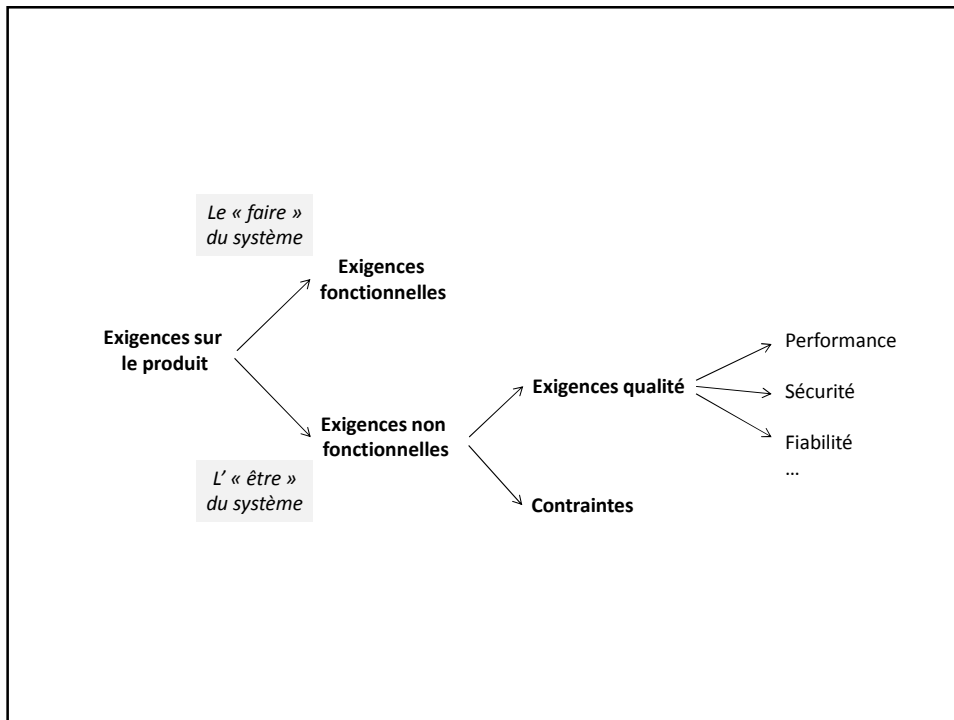
« Faites étudier par vos services, une voiture pouvant transporter deux cultivateurs en sabots, cinquante kilos de pomme de terre ou un tonnelet à une vitesse de 60 km/h, pour une consommation de 3 litres au cent. La voiture pourra passer dans les plus mauvais chemins; elle devra pouvoir être conduite par une conductrice débutante et avoir un confort irréprochable. Son prix devra être inférieur au tiers de la traction avant 11 CV. Le point de vue esthétique n'a aucune importance. »

97

## Exemples d' exigences non fonctionnelles

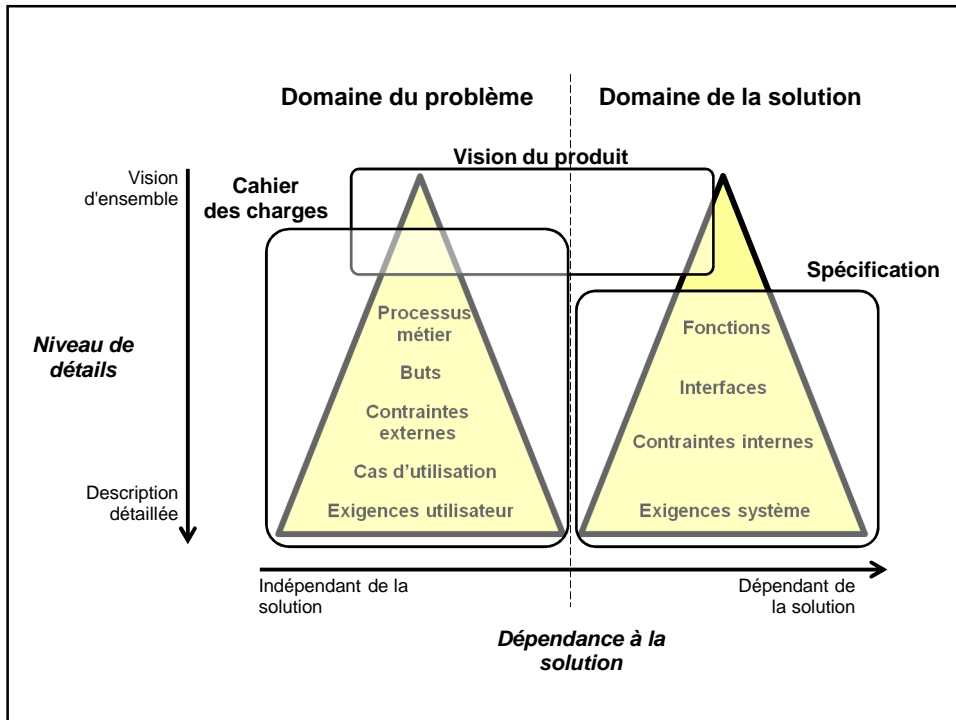
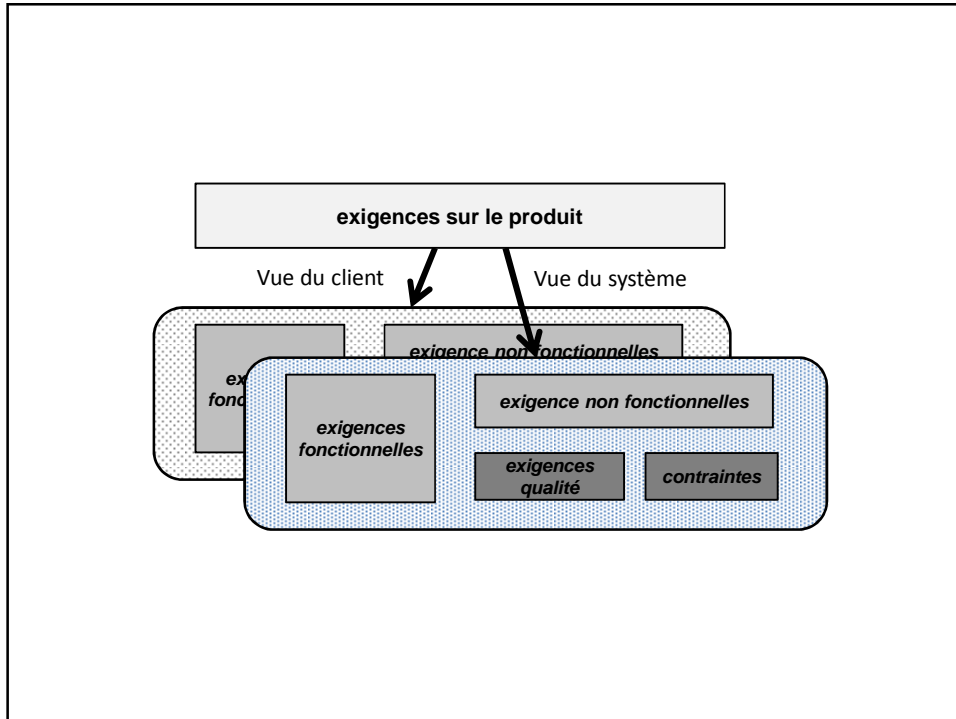
- Le temps d'apprentissage moyen de 80% des fonctionnalités du système doit être de moins de 3h
- La taille du programme ne doit pas dépasser 1Mbyte
- La fiabilité du système doit être telle que la probabilité d'erreur doit être inférieure à 0.0001
- A partir du système, il doit être possible d'utiliser ORACLE DBMS

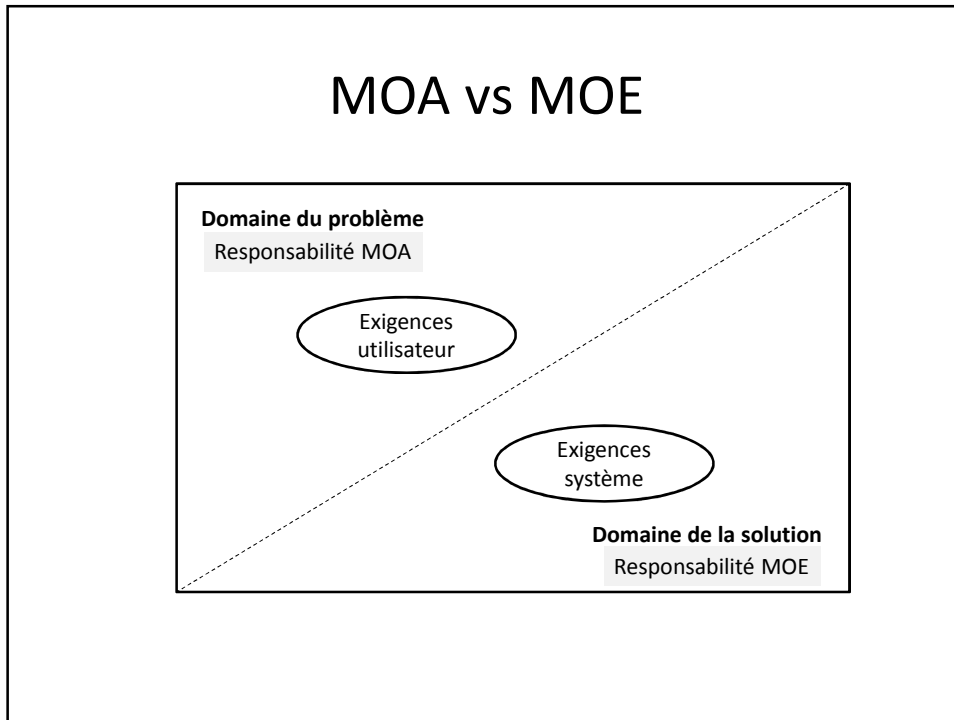
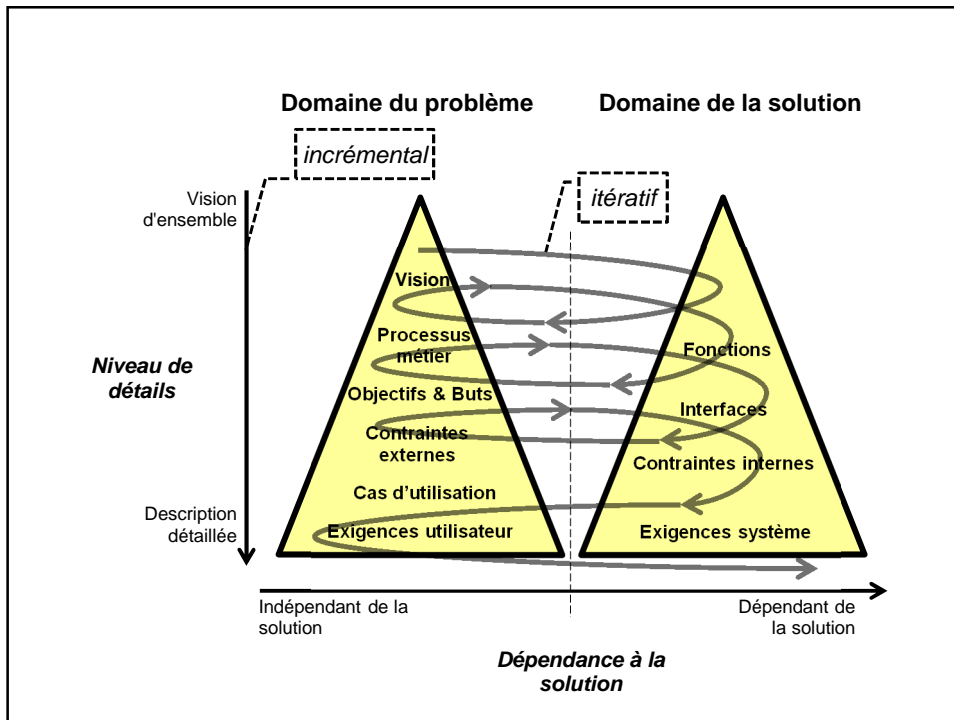
98



## Ingénierie des exigences

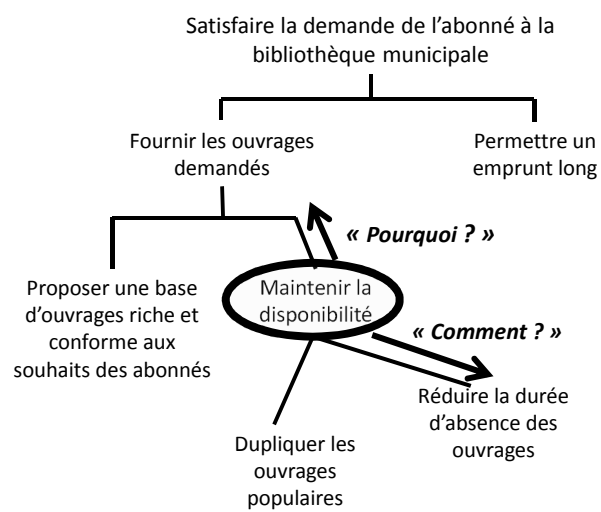
Utilisateur vs système



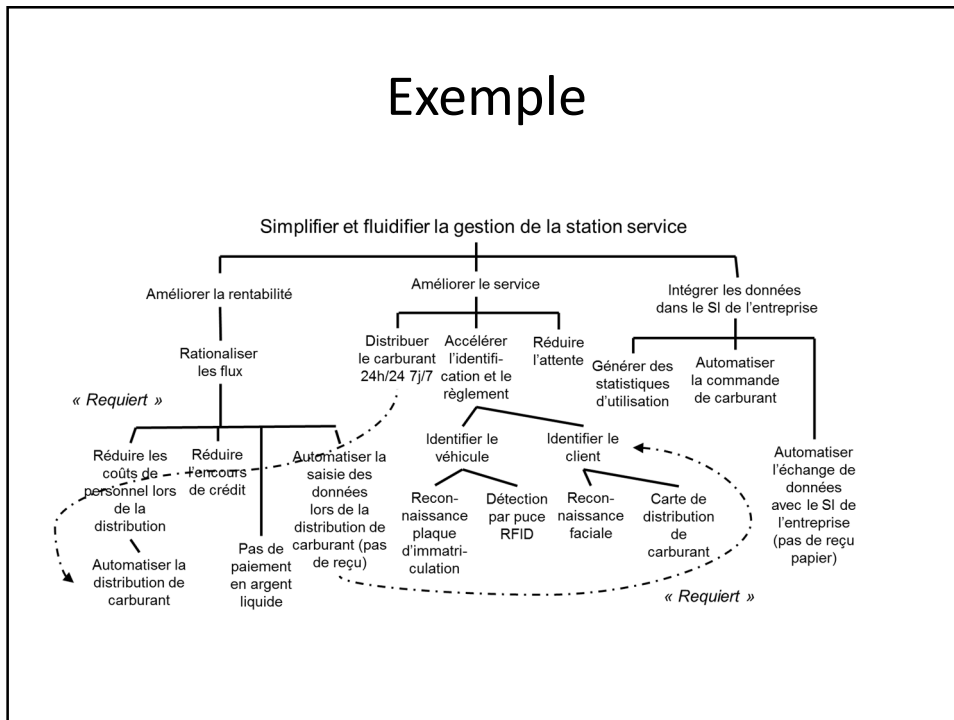


# Ingénierie des exigences

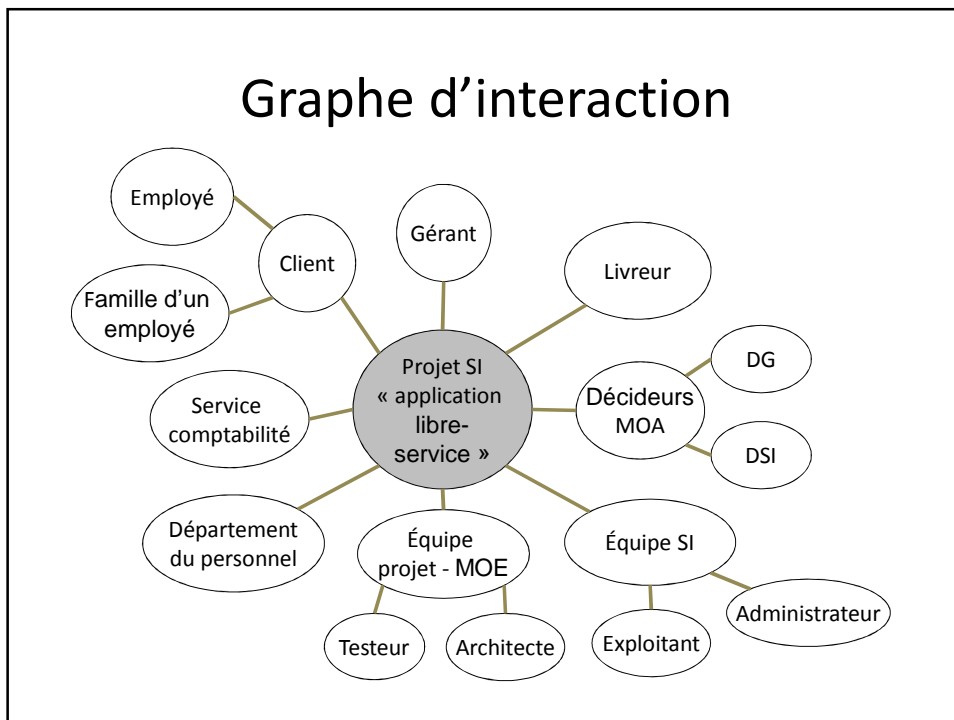
## Caractérisation d'une exigence

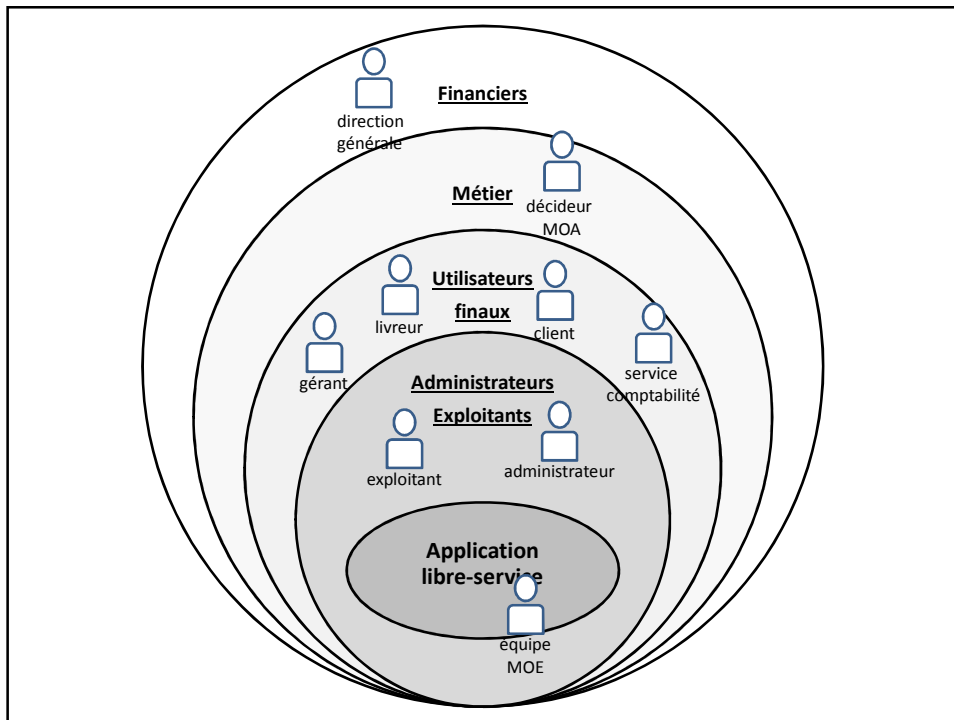


# Exemple



# Graphe d'interaction





## Ingénierie des exigences

Caractérisation d'une exigence



## Caractérisation d'une exigence

- Pour chaque exigence les critères suivants reviennent le plus souvent :
  - **atomique** : l'exigence est un élément identifiable et non décomposable ;
  - **concise** : lorsqu'elle est décrite en langage naturel, une exigence doit être écrite sous forme d'une seule phrase qui ne dépasse pas quelque ligne ;
  - **claire** : une lecture de l'exigence suffit pour la comprendre, la structure de la phrase est simple et n'utilise pas les subtilités littéraires ;
  - **précise** : tous les éléments utilisés dans l'exigence sont identifiable et complètement caractérisé (pas de question en suspend comme quelle est l'unité utilisée) ;
  - **abstraite** : une exigence ne doit pas imposer de solution (technique ou fonctionnelle) mais elle doit décrire le besoin ;
  - **non ambiguë** : la lecture de l'exigence doit permettre une compréhension du besoin, il est nécessaire de ne pas utiliser des tournures de phrase ou des mots permettant plusieurs interprétations ou complexifiant la compréhension de l'exigence ;
  - **complète** : il faut que l'ensemble des concepts utilisés dans l'exigence soit définit ;
  - **consistante** : il est nécessaire de vérifier que l'exigence forme un tout cohérent.
  
- D'autres critères peuvent être exigés ou nécessaire, ceux présentés ci-dessus constituent déjà une bonne base pour construire une méthodologie.



## Caractérisation d'un ensemble d'exigence

- Il est nécessaire qu'il n'y ait pas dans l'ensemble des exigences de redondance, la même information et/ou le même besoin apparaissant plusieurs fois.
- La cohérence d'un ensemble d'exigence concerne la bonne définition des concepts sur l'ensemble des exigences, dit autrement, il faut que chaque mot soit utilisé de la même façon pour chaque exigence.
- La complétude de l'ensemble des exigences est un point difficile, car elle est liée à la complétude de l'identification du besoin. Il est nécessaire lors de l'identification des exigences de vérifier que les exigences couvrent :
  - l'ensemble des objets manipulés,
  - l'ensemble des états des objets manipulés,
  - l'ensemble des conditions d'utilisation,
  - l'ensemble des scénarios d'exploitation qui ont été envisagés,
  - l'ensemble des normes applicables et des référentiels métier, etc.

## Caractérisation du processus

- Du point de vue processus, il faut que toute exigence soit :
  - **identifiable** : il est nécessaire d'associer à chaque exigence un identificateur unique (voir le premier attribut) ;
  - **vérifiable** : il est nécessaire de vérifier que toutes les exigences sont vérifiables. La vérification pouvant être une relecture, une modélisation, une analyse spécifique et/ou un test (voir le quatrième attribut).
  - **modifiable** : il faut être capable de gérer les évolutions des exigences durant toute la vie du système (réalisation, fabrication, mise en service, maintenance, retrait), pour ce faire, un processus de gestion de la configuration doit être mis en place (voir le 5<sup>ème</sup> attribut).

## Attributs caractérisant une exigence

Attributs	Description
ID	Identificateur unique
TEXTE	Texte de l'exigence
SOURCE	Elément ayant permis d'introduire cette exigence
VERIFICATION	Activité de vérification associée
VERSION	Version associée à l'exigence

## Exemple de collection d'attributs

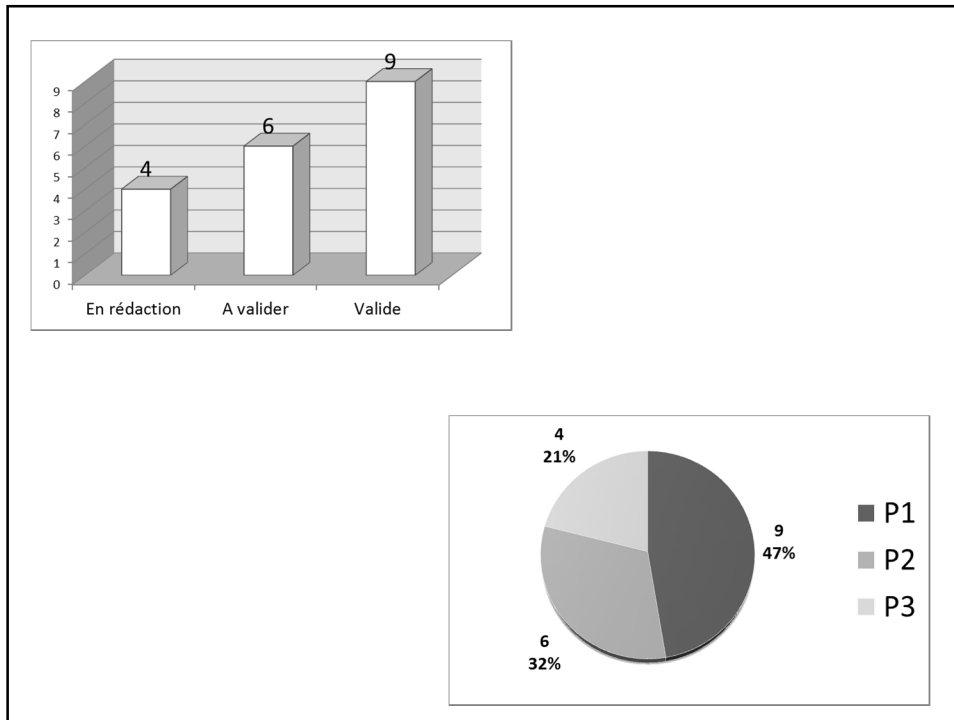
Attribut	Description	Exemple de valeurs
ID	Identifiant unique et invariant	1516
PROJET	Nom du projet qui introduit cette exigence	PROJET
TEXTE	Texte de l'exigence	« ... »
TITRE	Résumé de l'exigence	Gestion du temps
SOURCE	Elément ayant permis d'introduire cette exigence	Spécification système
CATEGORIE	Catégorie de l'exigence	Sécurité
AUTEUR	Auteur de l'exigence	Jules Martin
ETAT	Etat de l'exigence	Validée
CRITICITE	Criticité de l'exigence	Critique
PRIORITE	Priorité de l'exigence	P2
PORTEE	Portée contractuelle	Must
STABILITE	Stabilité de l'exigence dans le temps	Stable
VERIFICATION	Activité de vérification associée	Revue
VERSION	Version associée à l'exigence	1.3
EFFORT	Coût de l'exigence	Caillou
ALLOCATION	Allocation vers un élément de l'architecture	Sous-système RADIO
LOT	Lotissement de l'exigence	Lot 3

## Définition d'un attribut

Nom de l'attribut	Effort
Sémantique de l'attribut	Evaluation de l'effort (du coût) de l'implémentation de l'exigence
Liste de valeurs possibles	{sable, caillou, galet, rocher, falaise}
Sémantique des valeurs	<i>sable</i> : effort < 1H.J <i>caillou</i> : 1H.J <= effort < 5H.J <i>galet</i> : 5H.J <= effort < 20 H.J <i>rocher</i> : 20H.J <= effort < 200 H.J <i>falaise</i> : 200 H.J <= effort (où H.J = Homme / Jour)

## Complément

- Il est alors possible de compléter cette liste d'attributs par des attributs permettant de qualifier l'exigence:
  - Famille (fonctionnelle, FDM, sécurité, performance, ...);
  - Priorité (à définir en fonction du projet);
  - Vérifiable (Oui/Non);
  - Type de vérification (relecture, analyse spécifique, simulation, etc.);
  - Testable (Oui/Non);
  - Type de tests;
  - Source (Qui, quand, etc.);
  - État (à traiter, analysée, rejetée, etc.);
  - Type de document;
  - ...
  - Version.
- La définition des attributs est à réaliser en début de projet.



## Caractérisation de la spécification des exigences

- Il a été indiqué qu'un ensemble d'exigence doit être non redondant, cohérent et complet mais le document de spécification des exigences doit vérifier des critères complémentaires :
  - Le document doit être structuré ;
  - Le document doit être modulaire : les exigences qui vont ensemble sont regroupées dans une structuration claire, et sont proches les unes des autres ;
  - Le document doit être extensible (prise en compte de la maintenance) ;
  - Le document doit être suffisant : il n'est pas nécessaire de se replonger dans l'ensemble des documents sources pour comprendre le produit.

## Langage naturel

- La première règle consiste à utiliser et à respecter un gabarit d'exigence qui sera à définir pour le projet.
- Ce gabarit peut s'appuyer sur les règles suivantes (non-exhaustives) :
  - Une exigence doit être sous la forme « sujet + verbe + complément » ;
  - Utiliser de manière consistante le verbe « devoir », conjugué au présent ;
  - Utilisation de la forme active ;
  - Utilisation de termes définis de façon unique, ceci pouvant être mis en œuvre au travers de l'utilisation d'un glossaire (voir la section suivante) ;
  - Eviter l'utilisation des adverbes qui rendent ambiguës la phrase ;
  - Eviter les négations, leur utilisation doit être limitée aux exigences dites de sécurité (le système ne doit pas ...).

## Exemple

- Ci dessous quelques exemples d'exigence textuelle:

**[EXI\_1]**

Une arme a feu à pour objectif de propulser un balle dans une direction donnée.

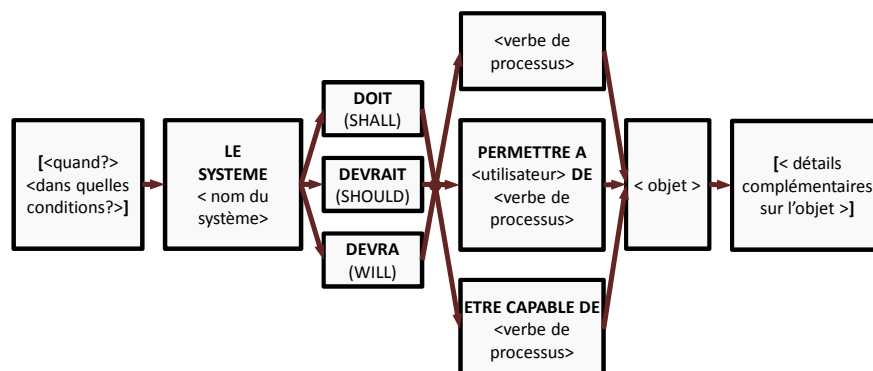
**[FIN\_EXI]**

**[EXI\_2]**

Une balle sera expulsée de la chambre de l'arme si l'arme n'est pas enrayée et si il y a mise à feu.

**[FIN\_EXI]**

## Exemple de schéma

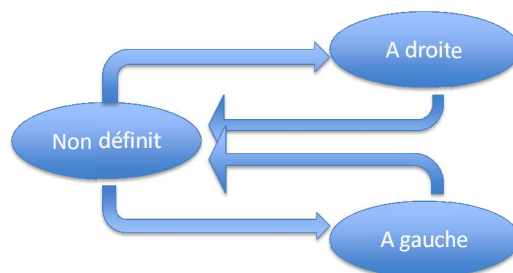


## Glossaire

- L'usage d'un glossaire doit répondre à des règles strictes :
  - Le glossaire doit être centralisé et versionné, c'est un référentiel ;
  - Les responsabilités relatives à la maintenance du glossaire doivent être définies ;
  - Le glossaire doit être maintenu et accessible tout au long du projet ;
  - L'utilisation du glossaire est obligatoire pour les parties prenantes ;
  - Le glossaire doit être approuvé par l'ensemble des parties prenantes.
- Il est recommandé de mettre en place le glossaire dès que possible, car toute incompréhension peut amener à des reprises mais surtout à des difficultés de compréhension qui seront découvertes tardivement et dont l'impact en effort et en coûts peut être très important.

## Exemple d'exigence

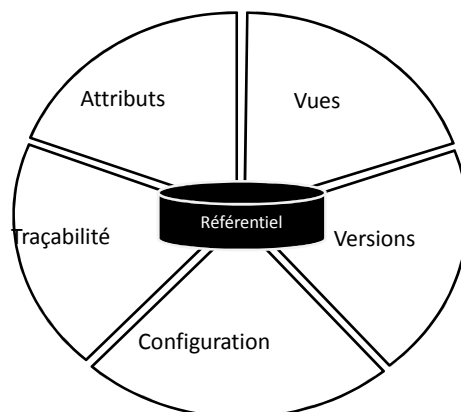
REQ\_0001 : Le comportement d'une aiguille doit respecter l'automate suivant



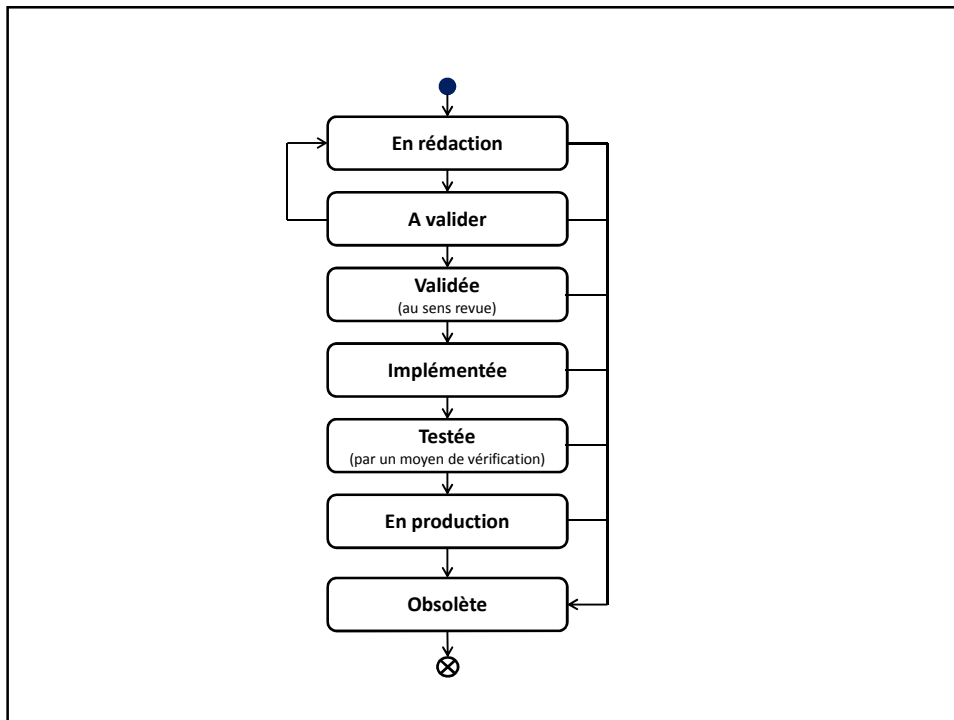
# Ingénierie des exigences

Cycle de vie d'une exigence

## Multi-vues



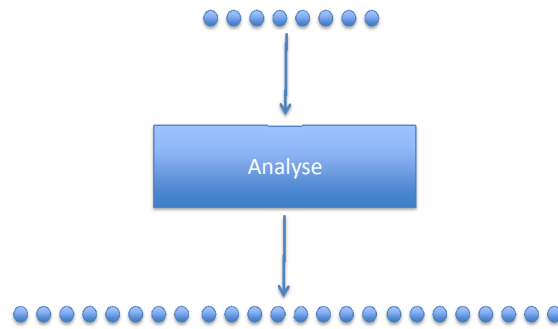




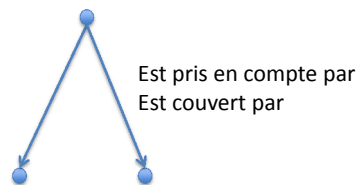
# Ingénierie des exigences

Traçabilité

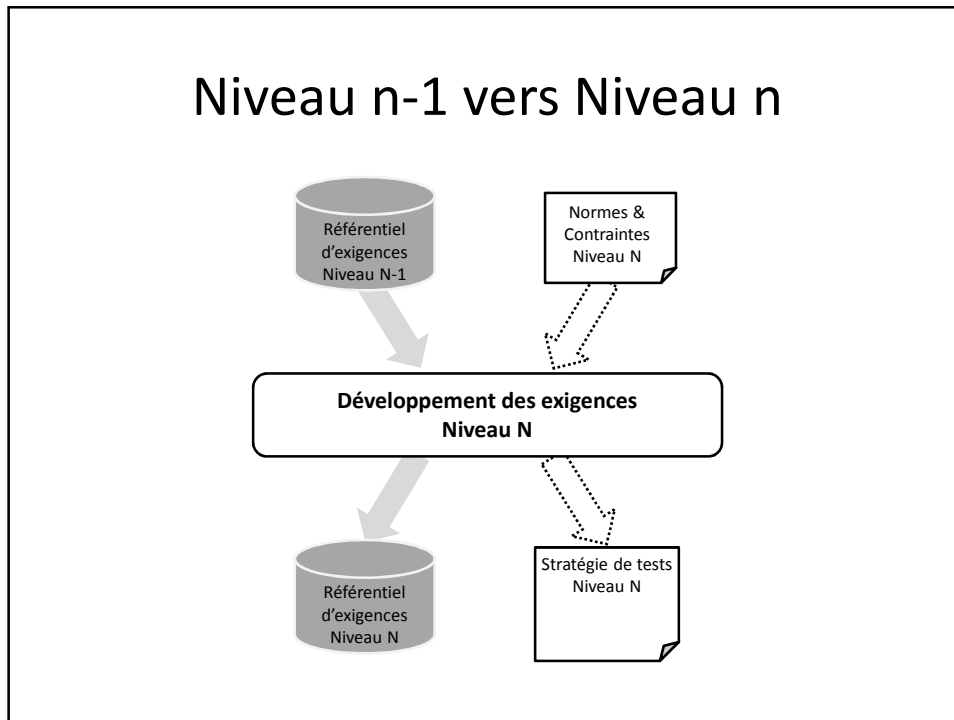
## Analyse - élicitation (1)



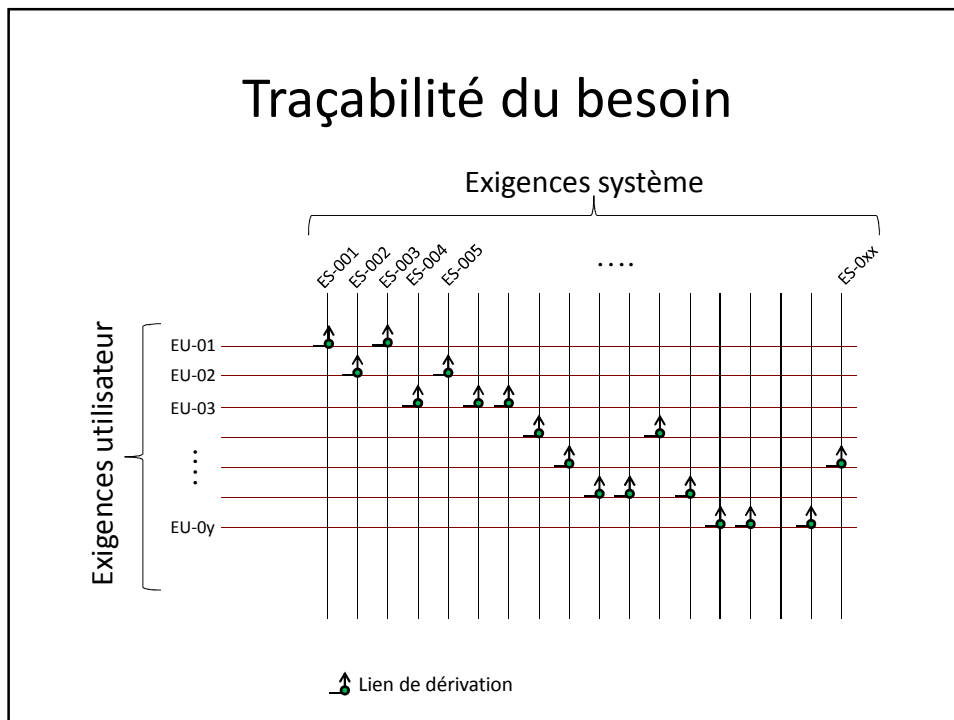
## Analyse - élicitation (2)



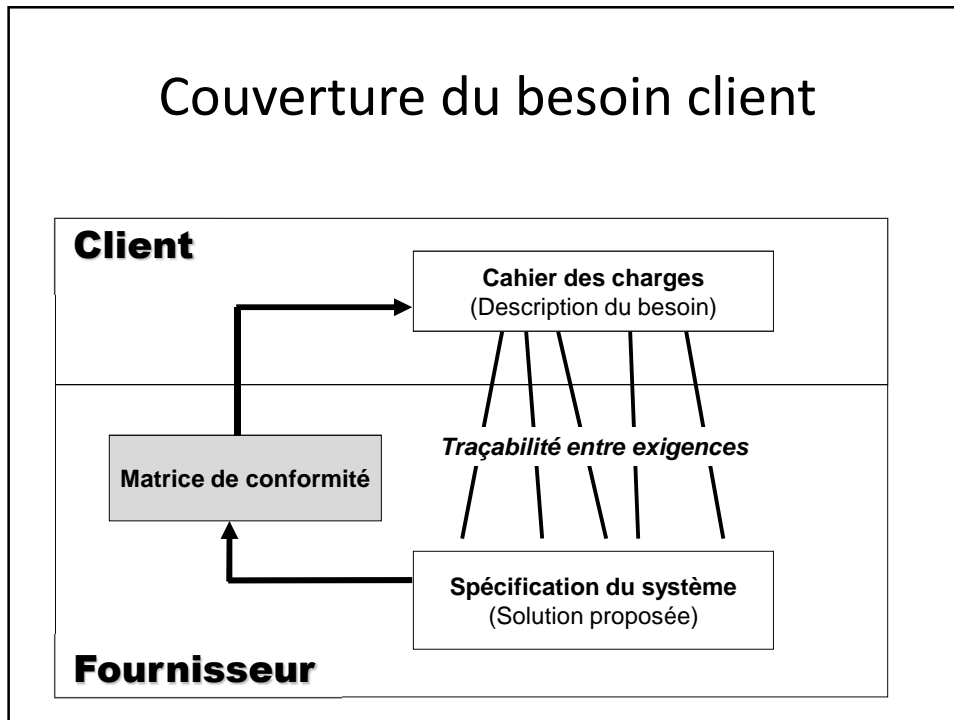
## Niveau n-1 vers Niveau n



## Traçabilité du besoin

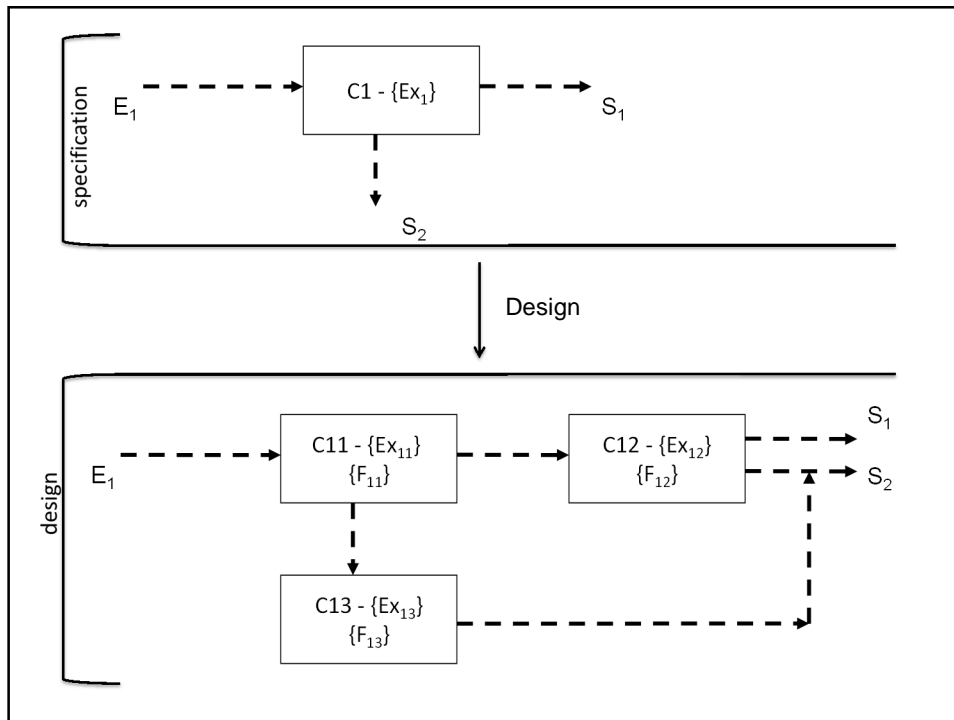


## Couverture du besoin client

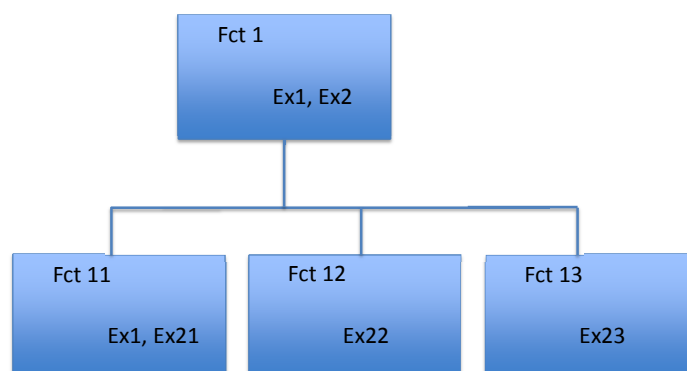


## Démonstration couverture du besoin

Exigence client	Titre	Couverture système	Justification
EXI-001	Identifier le véhicule	partielle	seule l'identification par plaque d'immatriculation est prévue
EXI-002	Identifier le véhicule par reconnaissance plaque d'immatriculation	oui	
EXI-003	Identifier le véhicule par détection d'une puce RFID	non	RFID non supporté
EXI-004	Distribuer du carburant 24h/24 et 7j/7	oui	
EXI-005	Réduire l'attente devant les pompes	oui	
EXI-006	Identifier le client	oui	
EXI-007	Identifier le client par une carte de distribution de carburant	oui	
EXI-008	Identifier le client par reconnaissance faciale dans 99% des cas	partielle	le pourcentage de détection est de 90% maximum
EXI-009	Améliorer le service rendu par la station service	oui	
...			



### Analyse - élicitation (3) Arborescence fonctionnelle



## Analyse - élicitation (5) matrice de traçabilité

	EX1	EX2
Fct11	EX1	EX21
FCT12		EX22
FCT13		Ex23

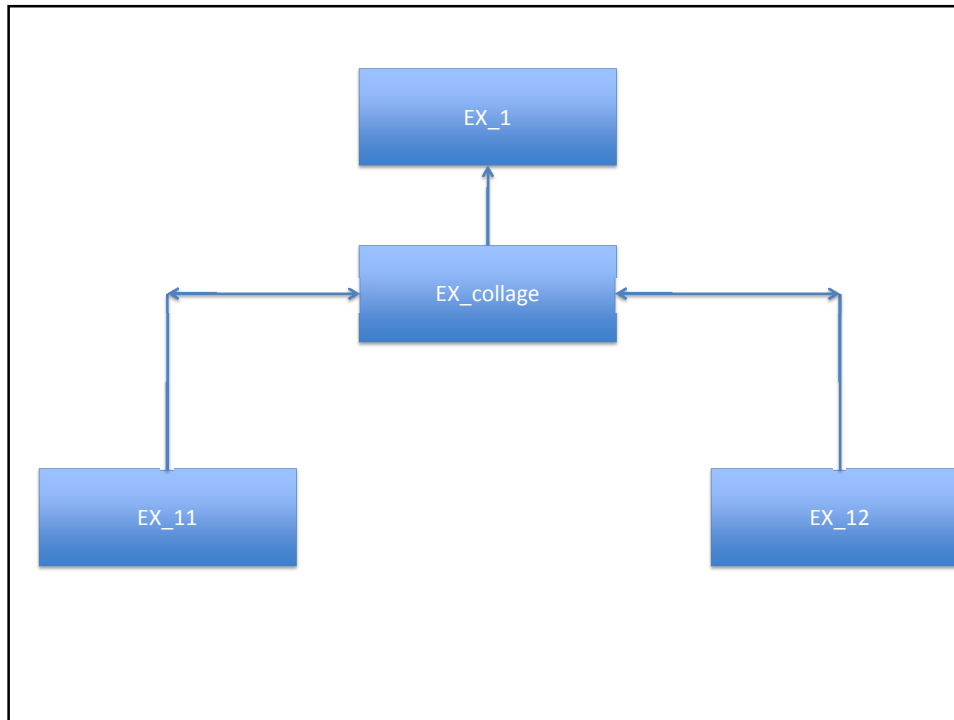
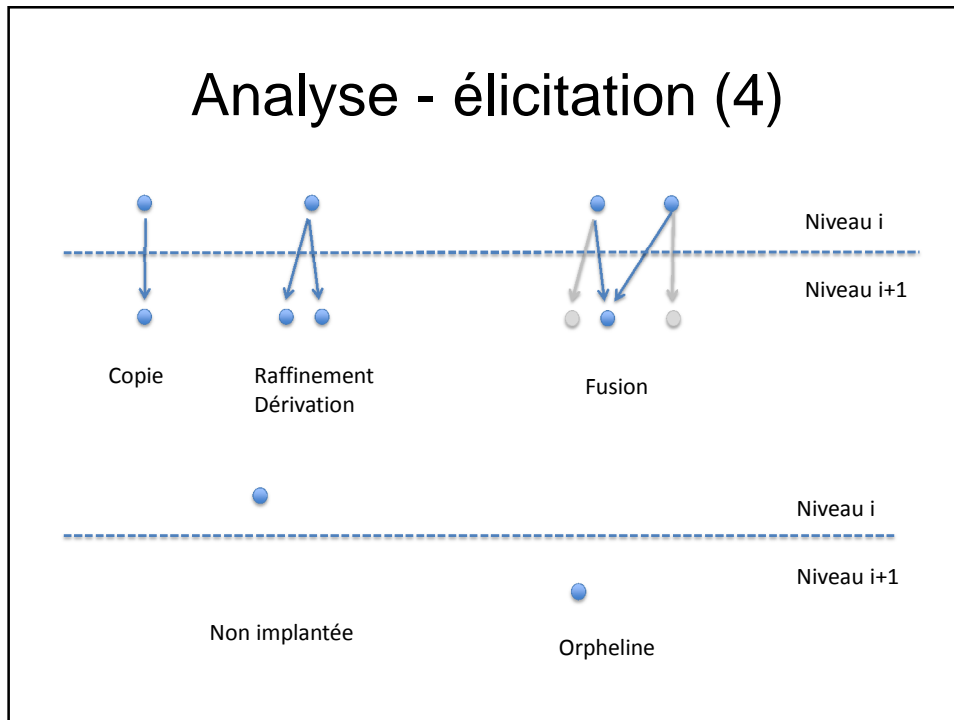
Niveau i	Niveau i+1
EX1	EX1
EX2	EX21, EX22, EX23

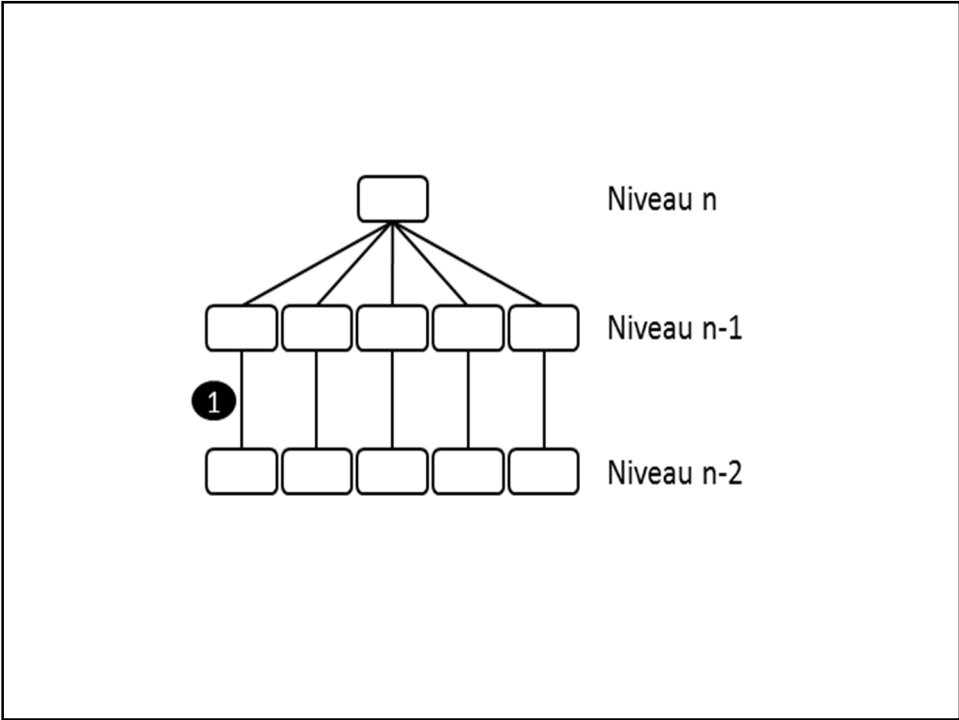
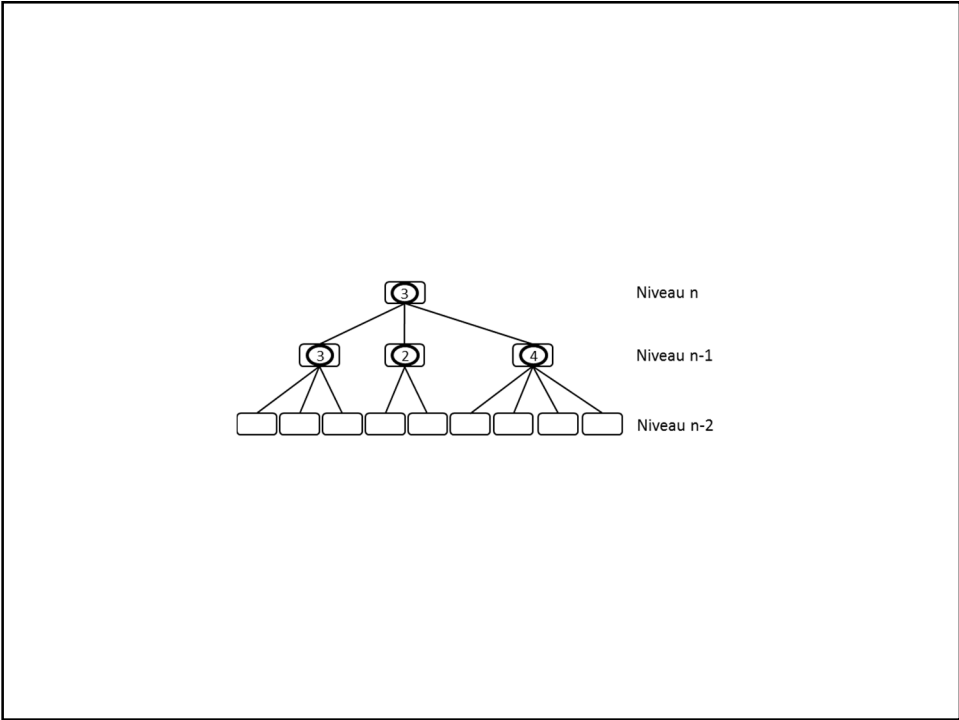
Niveau i+1	Niveau i
EX1	EX1
EX21	EX2
EX22	EX2
EX23	EX2

## Allocation

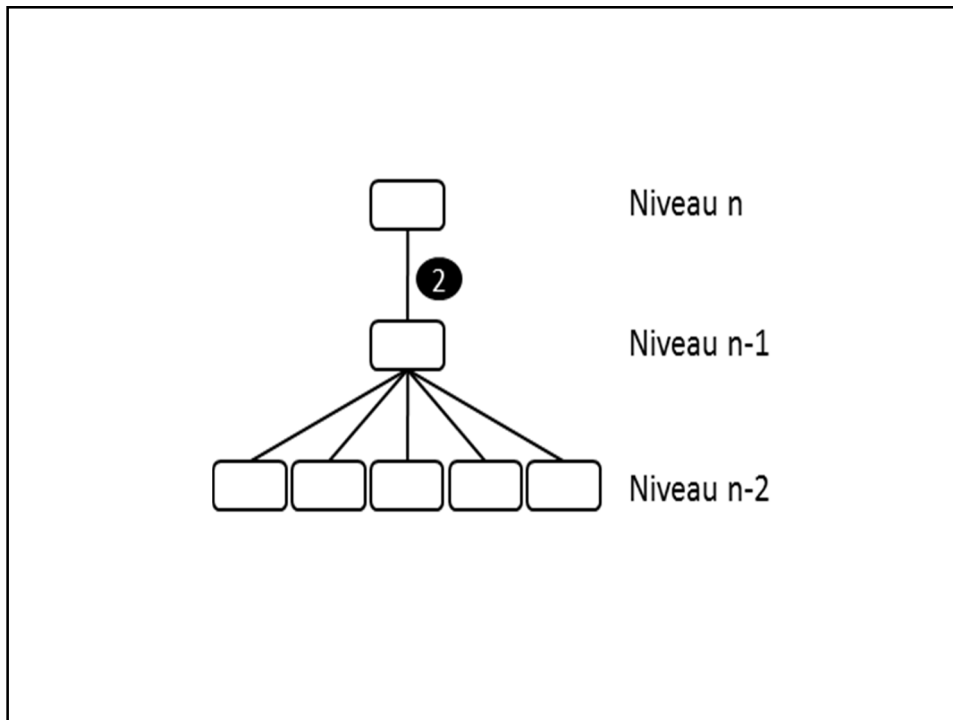
ID	TITRE	TEXTE	AUTEUR	SOURCE	CRITICITE	PRIORITE	STABILITE	ETAT	LOT	ALLOCATION
1	Fréquence	Le système doit ...	S. Steve	EdB §2	C1	P2	Evolutif	A valider	Lot 1	RADIO
	Réglage									
2	volume	Le système doit ...	A. Martin	Doc STB	C3	P1	Stable	En rédaction	Lot 1	RADIO
3	Basses	Le système doit ...	A. Martin	Doc SAG	C2	P1	Evolutif	A valider	Lot 2	AUDIO CD
	Réglage									
4	volume	Le système doit ...	S. Steve	EdB §2	C1	P3	Stable	Valide	Lot 1	RADIO
5	Fréquence	Le système doit ...	A. Martin	EdB §2	C1	P2	Stable	Valide	Lot 1	RADIO
6	MP3	Le système doit ...	G. Simon	Doc SAG	C3	P2	Evolutif	En rédaction	Lot 2	AUDIO CD
	Basses									
7	affichage	Le système doit ...	G. Simon	Doc STB	C2	P3	Instable	A valider	Lot 2	AUDIO CD
8	Fréquence	Le système doit ...	S. Steve	Doc SAG	C2	P1	Instable	A valider	Lot 1	RADIO
9	Fréquence	Le système doit ...	A. Martin	EdB §3	C3	P2	Stable	A valider	Lot 1	RADIO
	Réglage									
10	volume	Le système doit ...	G. Simon	Doc STB	C1	P1	Instable	Valide	Lot 2	RADIO
	Réglage									
11	volume	Le système doit ...	G. Simon	Doc STB	C1	P1	Instable	Valide	Lot 2	RADIO
12	Basses	Le système doit ...	G. Simon	EdB §4	C3	P1	Stable	Valide	Lot 2	AUDIO CD
13	Aigus	Le système doit ...	A. Martin	Doc STB	C1	P1	Stable	Valide	Lot 2	AUDIO CD
14	DTS-2	Le système doit ...	S. Steve	Doc STB	C1	P3	Evolutif	En rédaction	Lot 2	AUDIO CD
15	Fréquence	Le système doit ...	S. Steve	Doc STB	C3	P2	Instable	Valide	Lot 1	RADIO
16	Aigus	Le système doit ...	S. Steve	EdB §2	C2	P2	Stable	En rédaction	Lot 1	AUDIO CD
	Réglage									
17	volume	Le système doit ...	S. Steve	EdB	C2	P1	Stable	Valide	Lot 2	RADIO
18	DTS	Le système doit ...	S. Steve	Doc SAG	C2	P1	Stable	Valide	Lot 2	AUDIO CD
19	MP3-3	Le système doit ...	G. Simon	EdB §4	C2	P3	Stable	A valider	Lot 2	AUDIO CD

## Analyse - élicitation (4)



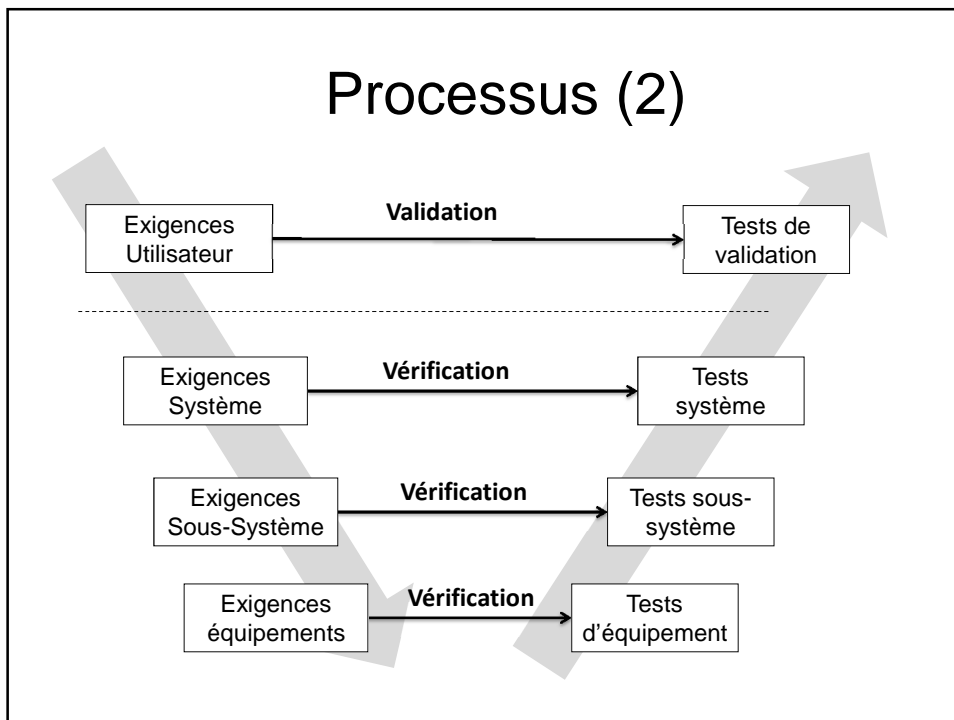
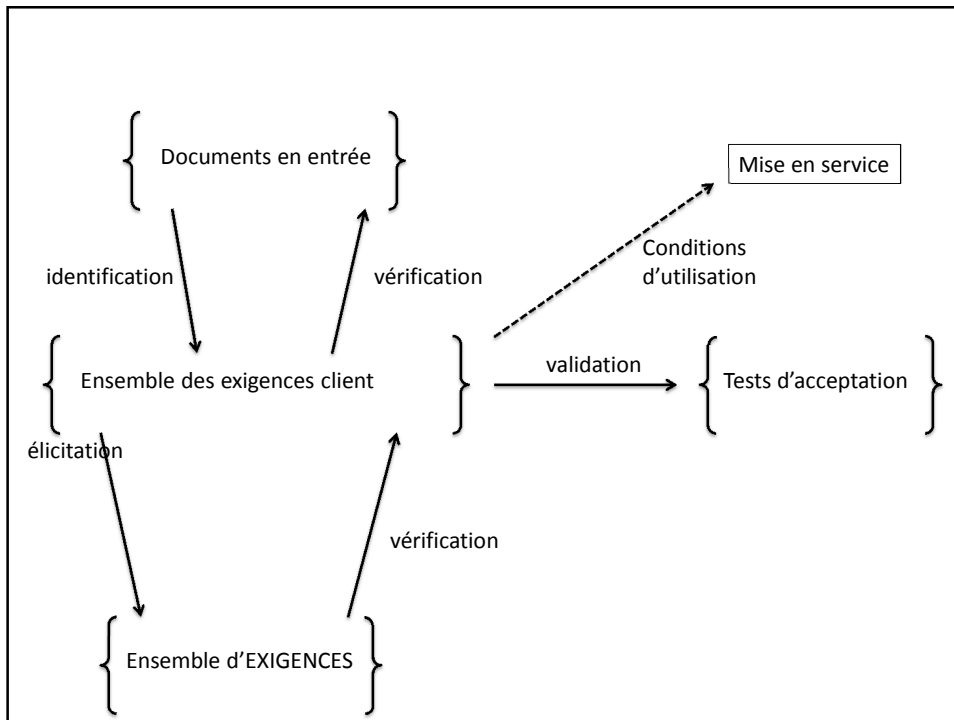




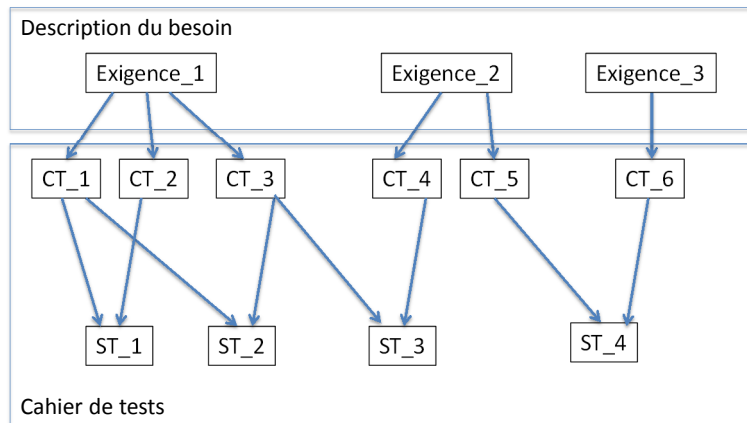


## Ingénierie des exigences

Savoir en évaluer la testabilité



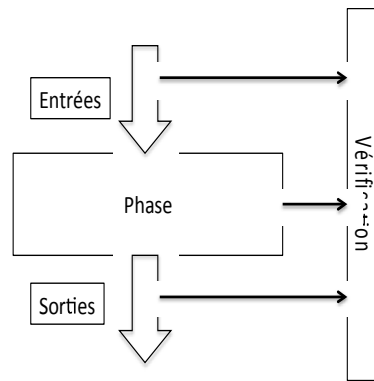
## Stratégie de test



## Ingénierie des exigences

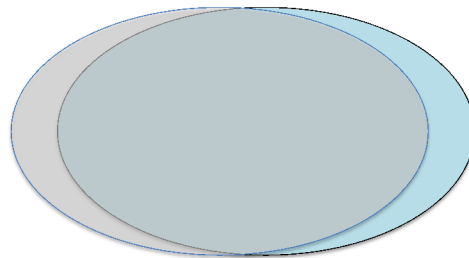
Savoir mettre en œuvre une revue  
d'exigence

## Vérification



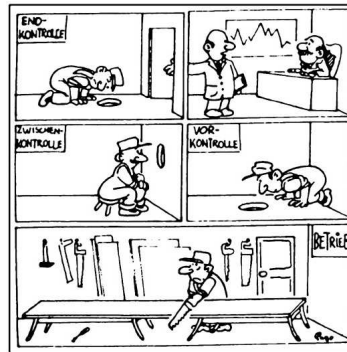
## Pourquoi vérifier

Fonctions Codées



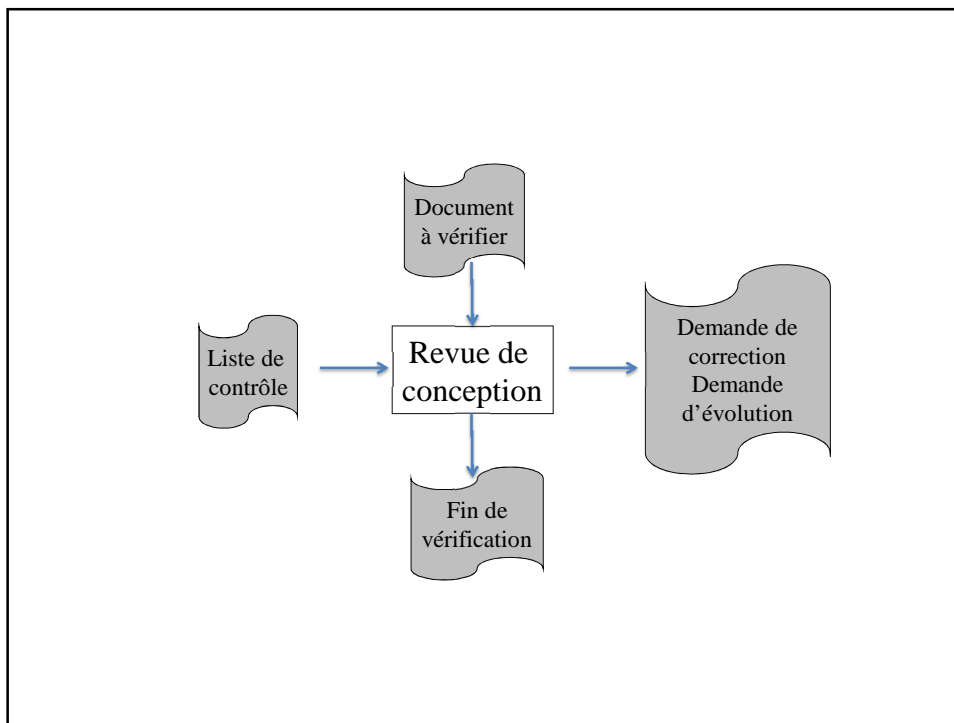
Fonctions spécifiées

## Vérification de la vérification



- Pour chaque phase du processus de réalisation, il faudra mettre en place une phase de vérification des exigences qui a trois objectifs :
  - Que le besoin initial a été pris en compte : une traçabilité entre les exigences du niveau supérieur et les exigences de ce niveau doit être mise en place. Cette traçabilité doit être vérifiée: les liens existent et ils sont justifiés ;
  - Que l'ensemble d'exigences forme un tout qui est correct : il faut démontrer que les exigences sont compréhensibles, non-ambiguës, vérifiables, faisables, ... Il faut démontrer que l'ensemble des exigences est complet et cohérent (pas de conflit) ;
  - Qu'aucun élément non traçable n'a été introduit : le but de cette vérification est de contrôler que toutes les exigences en cours de rédaction sont traçable avec un besoin du niveau supérieur. Très souvent, on voit apparaître des exigences qui sont dites exigences de conception et/ou d'architecture qui n'ont aucun lien avec le niveau supérieur. Une analyse de ces exigences montre qu'il s'agit pour la plus part du temps de non dit et très rarement d'exigence de conception. Il faut d'ailleurs définir ce qu'est une exigence de conception et/ou d'architecture.

DAL	DCP
DAL_EX_1	DCP_EX_11, DCP_EX_12, DCP_EX_13
DAL_EX_2	
DAL_EX_3	DCP_EX_11



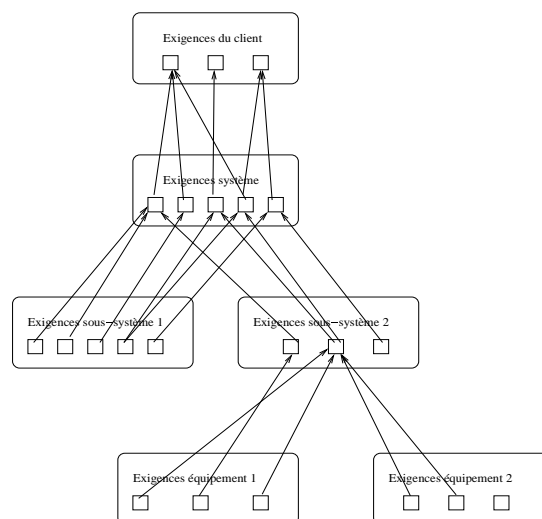
Point	Règle	Etat OK/KO	Commentaire
R_1	Toutes les exigences mères doivent être tracées ou une justification doit être donnée		
R_2	Toutes les exigences du document doivent être tracées avec au moins une exigence mère ou une justification doit être donnée		
R_3	Toutes les interfaces identifiées participent à au moins une exigence		
R_4	Tous les états du système participent à au moins une exigence		
R_5			
R_6	Chaque exigence est-elle atomique (pas de nécessité de lire un ensemble d'exigence pour comprendre le besoin)		
R_7	Chaque exigence est-elle vérifiable ?		
R_8			

## Ingénierie des exigences

Comprendre les concepts et l'utilité  
de la gestion des exigences

- Il existe deux types de projet,
  - pour certain projet, la phase d'acquisition des exigences est réalisée par le donneur d'ordre et fait parti d'un processus fortement maitrisé, dans ce contexte le référentiel en sortie de la phase d'acquisition des exigences est assez stable et une évolution de ce référentiel donnera lieu à une évolution du contrat ;
  - pour la seconde famille de projet, la phase d'acquisition fait partie du projet ou la réponse à appel d'offre a été élaborée avant que la phase d'acquisition est été finalisée. Dans ce cas la, l'ensemble des exigences n'est pas connu et peut évoluer durant un certain temps. Il est alors nécessaire de disposer d'un processus de gestion des changements.

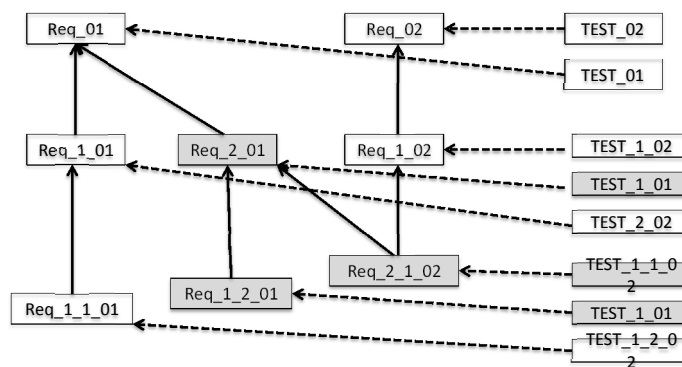
## Traçabilité



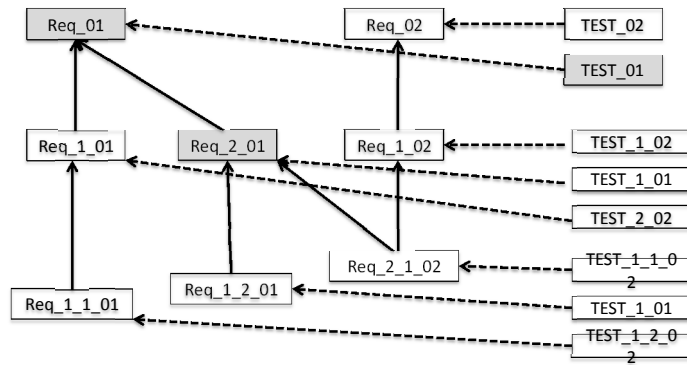


- L'analyse des changements est réalisée au travers d'une analyse d'impact (définition 1) et d'une analyse de non-régression (définition 2). Dans certain cas la non-régression est dite totale, et pour cela il est alors nécessaire de ré-exécuter l'ensemble des essais d'une ou de toutes les phases. L'analyse de non-régression à pour objectif de minimiser le cout d'une nouvelle version.
- **Définition 1** Analyse d'Impact  
L'analyse d'impact d'une anomalie consiste à identifier les modifications à réaliser sur la phase descendante (impact sur les documents, impact sur le code, impact sur la description et l'implémentations des tests) de la réalisation.
- **Définition 2** Analyse de Non Régression  
L'analyse de non-régression consiste à déterminer un ensemble de tests permettant de démontrer que la modification qui a été réalisée n'a pas d'effet sur le reste de l'application logicielle.
- Il est à noter qu'une analyse de non régression peut être réalisée sur l'application logicielle ou sur un élément plus important comme un équipement, un sous-système et/ou un système.

## Analyse d'impact

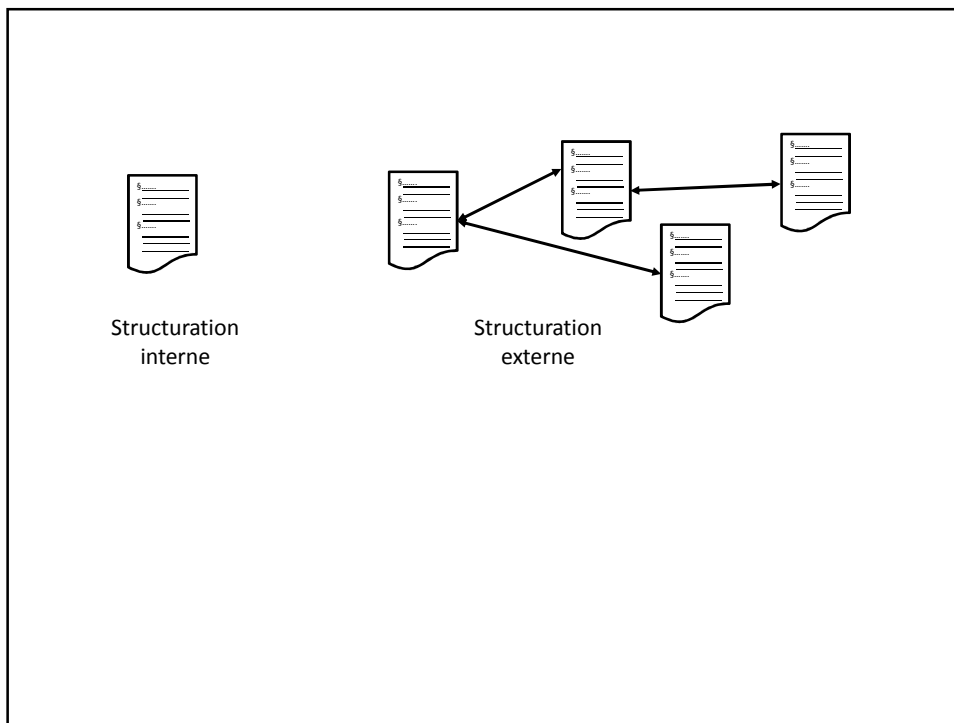
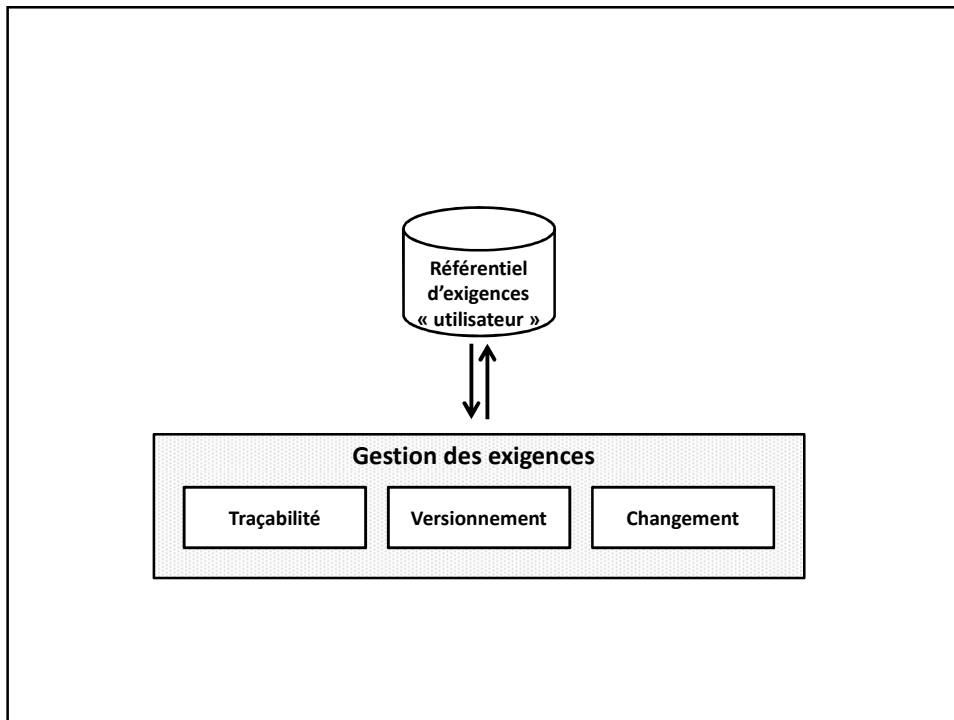


## Non-régression



## Ingénierie des exigences

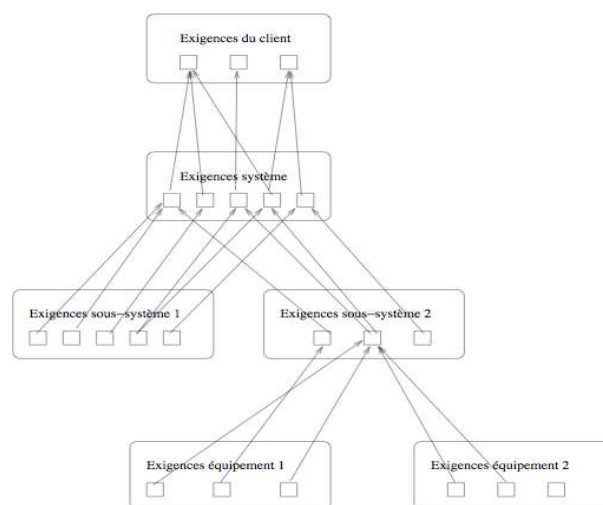
Savoir gérer le référentiel des exigences tout au long du projet



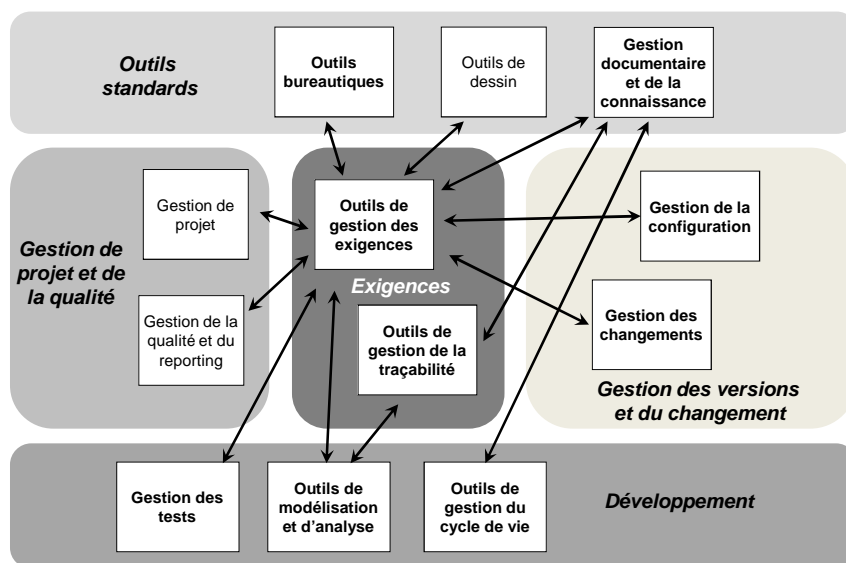
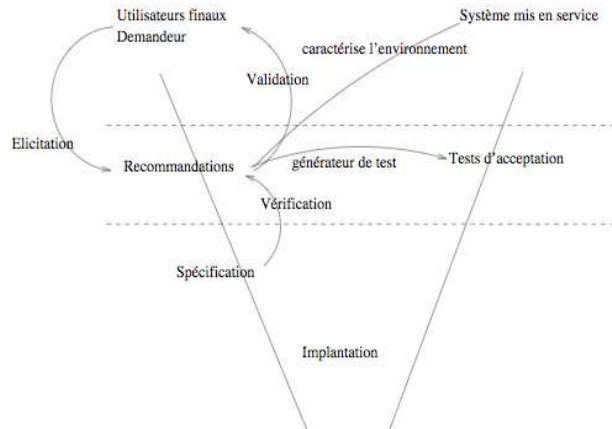
## 3 activités

- Processus :
  - définition, mise en œuvre et accompagnement d'un processus d'ingénierie des exigences (analyse, formalisation, évolution des exigences)
- Outils :
  - support, intégration des outils dans le processus de gestion des exigences
- Traçabilité :
  - définition et mise en œuvre de modèles de données, liens processus/outils...

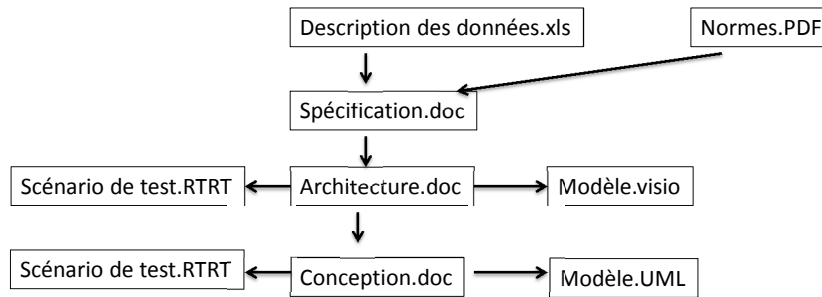
## Tout au long du projet



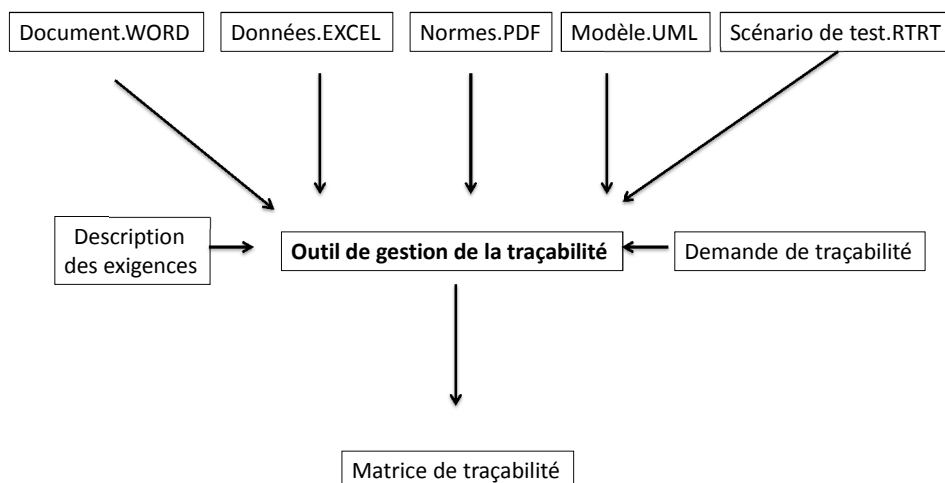
# Processus



## Processus de développement



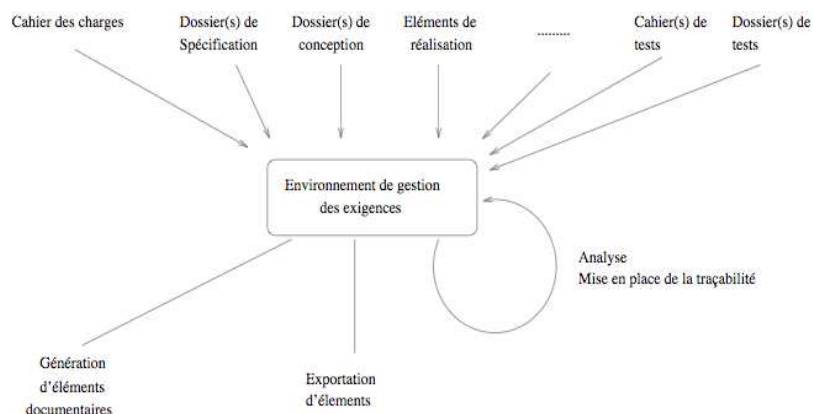
## Exigence et traçabilité



## Apports d'un outil

- Permet l'analyse des exigences;
- Assure la traçabilité des exigences
  - Génération des matrices de traçabilité;
- Permet la génération de tout ou partie de la documentation;
- Constitue un référentiel vivant consultable;
- Assiste dans le suivi des modifications;
- Fournit les données statistiques pour le projet;
- Permet le REX pour les futurs projets.

## Modèle central



## BILAN - IE

- L'ingénierie des exigences est un processus de l'ingénierie système.
- Il existe une littérature, des outils et des utilisations (utilisateurs).
- L'IE doit être mis en place sur l'ensemble du cycle de réalisation.

## Outils

- RTM d'Integrated Chipware Inc.
- DOORS de Telelogic/IBM
- RequisitePro de Rational
- Rectify de DASSAULT
- TRAMWAY (outil libre)
- DAWN
- ...



## Problématique/Difficulté

- **Savoir Exiger**
- Faire face à la complexité des systèmes donc à l'éparpillement des exigences et au nombre de documents
- Vérifier la cohérence et la complétude des exigences
- Suivre les exigences
- Améliorer le processus d'ingénierie

Ingénierie des exigences

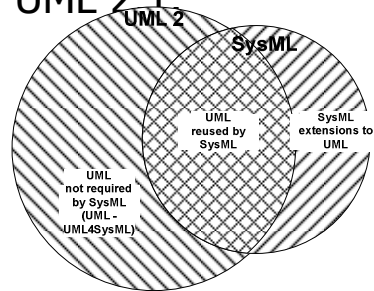
SysML

# SysML (1)

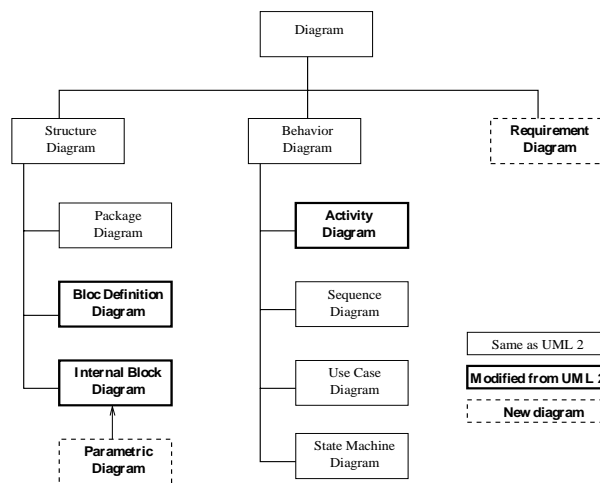
- SysML (\*) peut être vu comme un profil d'utilisation, mais il se définit lui-même comme une extension de la spécification de la superstructure UML 2.1

OMG Systems Modeling Language (OMG SysML™) Specification,  
OMG Final Adopted Specification, May 2006, ptc/06-05-04.

<http://www.sysml.org/>



# SysML (2)

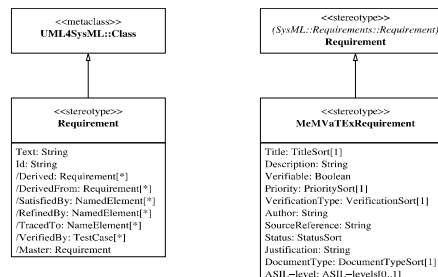


## SysML (3)

- SysML prend en compte le traitement des exigences au travers de deux nouveaux diagrammes nommés
  - le « diagramme d'exigence »
  - le « diagramme des paramètres »
- SysML propose les notions de « Requirement » et la notion de « TestCase » et un ensemble de quatre liens :
  - la composition,
  - la dérivation,
  - la satisfaction et
  - la vérification.

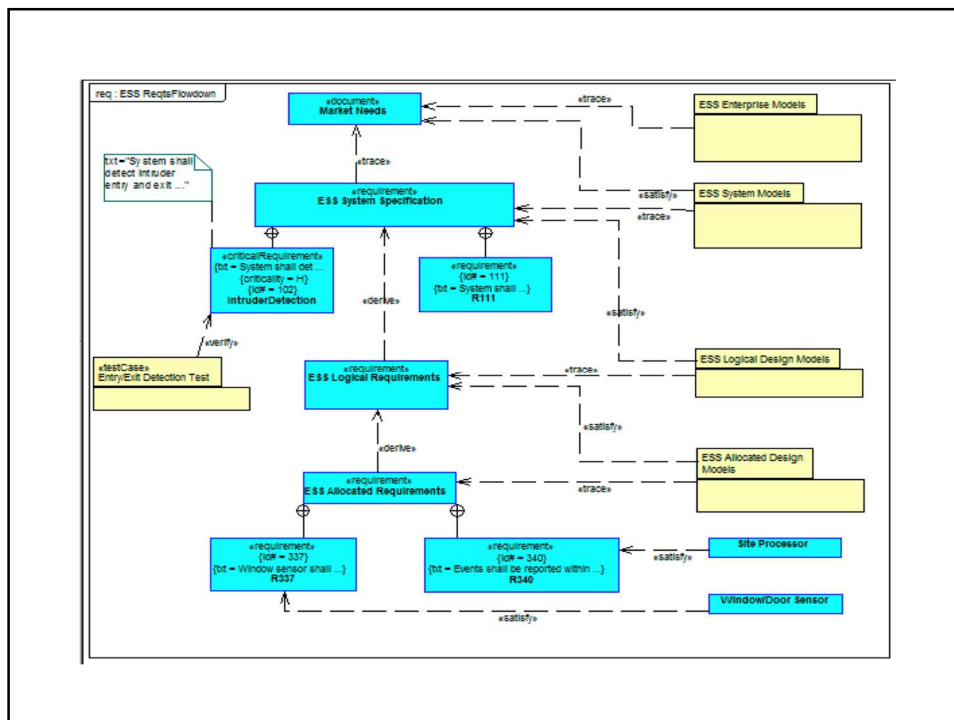
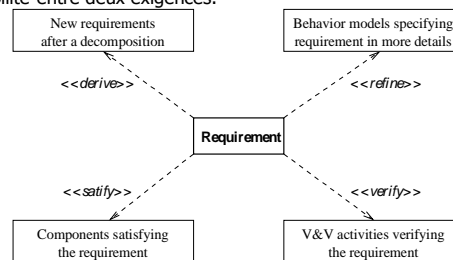
## Diagramme d'exigence (1)

- Le diagramme des exigences permet de modéliser les exigences, les cas de test et l'ensemble des liens (raffinement, trace, vérification, ...). Les exigences peuvent ensuite être liées avec les éléments des autres diagrammes au travers de lien « callout ».
- Une exigence est au minimum composée de trois attributs (properties) :
  - le nom,
  - l'identificateur et
  - le texte.

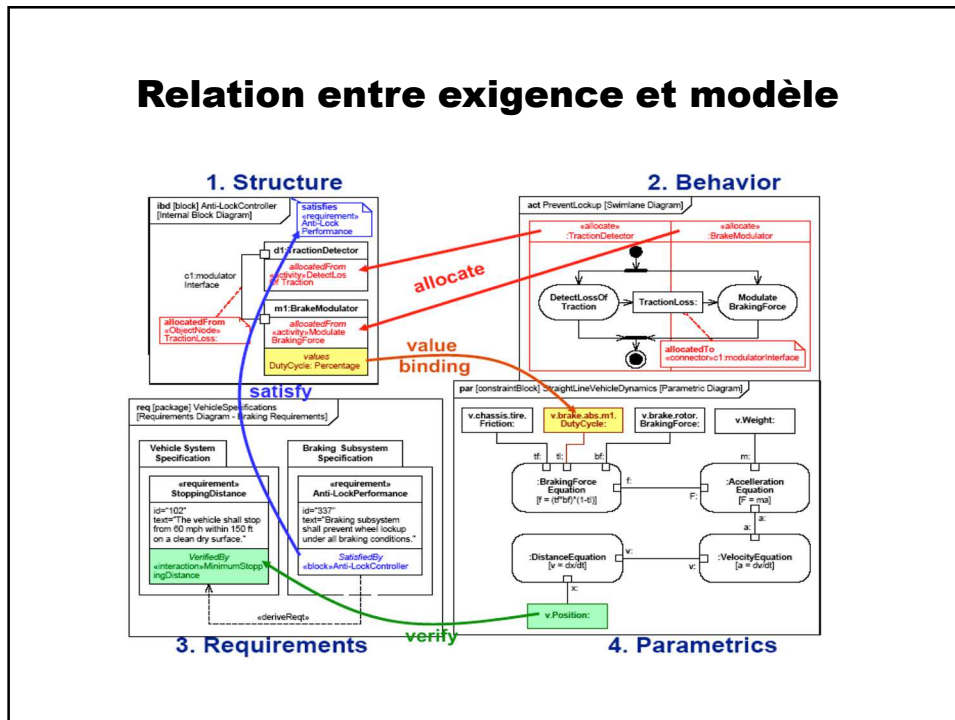


## Diagramme d'exigence (2)

- Il existe des liens qui permettent de faire le lien entre les exigences :
  - Containment : une exigence est décomposée en plusieurs exigences ;
  - Copy : L'exigence du fournisseur est la reprise à l'identique de l'exigence du client ;
  - DeriveReq: indique que l'exigence est élaborée à partir d'une exigence plus générale ;
  - Verify : indique le lien entre une exigence et un cas de test et
  - Satisfy : indique qu'une exigence est prise en compte par un élément du modèle.
  - Refine : une exigence est précisée par une autre ;
  - Trace : il y a un lien de traçabilité entre deux exigences.



## Relation entre exigence et modèle



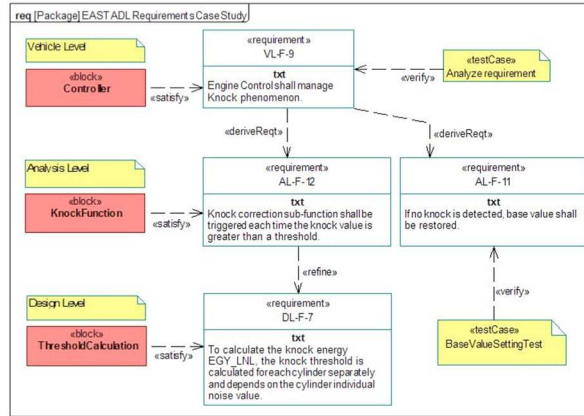
## Génération de table de traçabilité

id	name	text
2	Performance	The Hybrid SUV shall have the braking, acceleration, and off-road capability of a typical SUV, but have dramatically better fuel economy.
2.1	Braking	The Hybrid SUV shall have the braking capability of a typical SUV.
2.2	FuelEconomy	The Hybrid SUV shall have dramatically better fuel economy than a typical SUV.
2.3	OffRoadCapability	The Hybrid SUV shall have the off-road capability of a typical SUV.
2.4	Acceleration	The Hybrid SUV shall have the acceleration of a typical SUV.

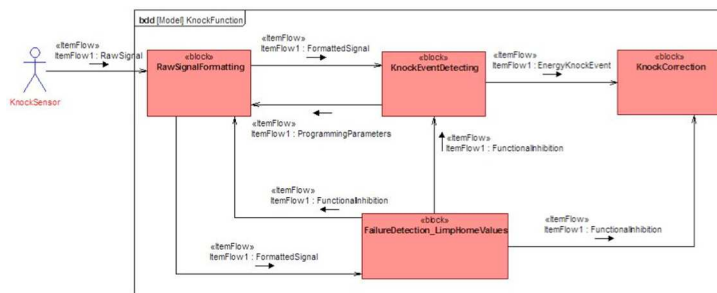
id	name	relation	id	name	relation	id	name
2.1	Braking	deriveReq	d.1	RegenerativeBraking			
2.2	FuelEconomy	deriveReq	d.1	RegenerativeBraking			
4.2	FuelCapacity	deriveReq	d.2	Range			
2.3	OffRoadCapability	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement
2.4	Acceleration	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement
4.1	CargoCapacity	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement

Figure B-14. Requirements Relationships Expressed in Tabular Format (Table)

# Exemple (1)



# Exemple (2)



Name	Txt	Derived	Derived From	Refines	Refined By	Satisfied By	Verified By
VL-F-9	Engine Control shall manage Knock phenomenon.		«requirements» AL-F-12 (Requirements: AL) «requirements» AL-F-11 (Requirements: AL)			«block» Controller (Requirements)	«Use Case» Analyze requirement (Requirements)
AL-F-11	If no knock is detected, base value shall be restored.	«requirements» VL-F-9 (Requirements: VL)					«Activity Diagram» BaseValueSettingTest (Requirements) «Operations» BaseValueSetting (Testing: BaseValueSetting)
AL-F-12	Knock correction sub-function shall be triggered each time the knock value is greater than a threshold.	«requirements» VL-F-9 (Requirements: VL)		«requirements» DL-F-7 (Requirements: DL)		«block» KnockFunction (Requirements)	
DL-F-7	To calculate the knock energy EGY_LNL, the knock threshold is calculated for each cylinder separately and depends on the cylinder individual noise value.				«requirements» AL-F-12 (Requirements: AL)	«block» ThresholdCalculation (Requirements)	

## Conclusion

- SysML est un standard de l'industrie;
- Les outils le supporte plus ou moins complètement.
- Les industriels en ont besoin.
  
- Il y a des liens/passerelles entre les outils classiques et les outils UML/SysML.
  - Export/IMPORT liste exigences vers DOORS, WORD, ..
  - Export/Import liens entre exigences ...
  - ...

Ingénierie des exigences

Certification

