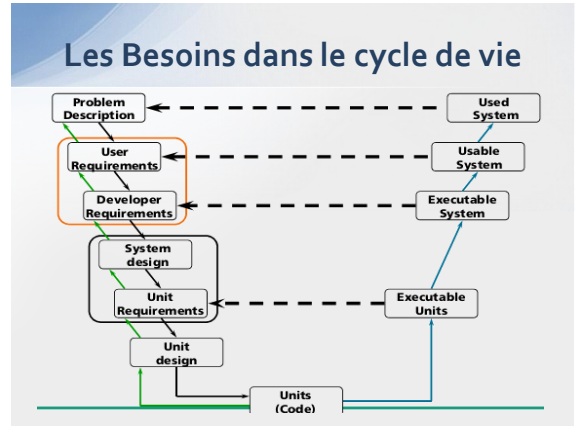
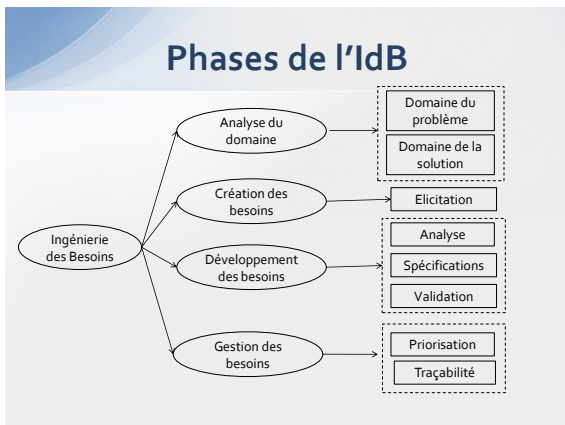
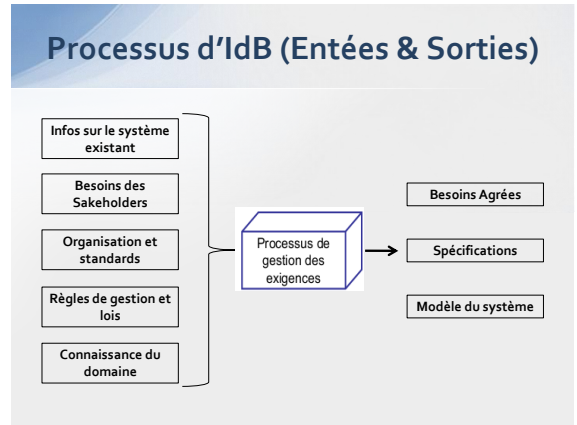


Département Informatique
 Master1 SIGL
 Cours: Ingénierie des Besoins
 Chapitre 1
 Processus d'Ingénierie
 des Besoins



Définition du Processus d'IdB

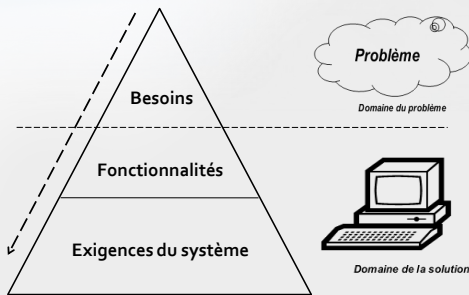
- « Étude systématique des besoins d'un utilisateur, visant à définir un système.»
 Office de la langue française, 2000
- « The systematic process of developing requirements through an iterative cooperative process of analysing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained.»
 K. Pohl, 1993



Phases de l'IdB

- Début du processus** (vision, besoin ou opportunité, bonne idée...), étude de faisabilité, étendue du système, risques, etc.
- Phase de création des besoins**
 - Élicitation (élucidation) des besoins**: Les besoins sont découverts en consultant les diverses parties prenantes.
- Phase de développement des besoins**
 - Analyse des besoins et négociation**: Les besoins sont analysés et les conflits résolus, souvent par négociation.
 - Spécification des besoins**: Un document précis décrivant les besoins est créé.
 - Validation des besoins**: La spécification des besoins est vérifiée en termes de cohérence et de complétude.
- Phase de Gestion des besoins**
 - Priorisation**: identifier, contrôler et définir les priorités des besoins
 - Traçabilité**: tracer l'évolution des besoins et suivre les modifications qui leur sont apportées.

Gestion des besoins



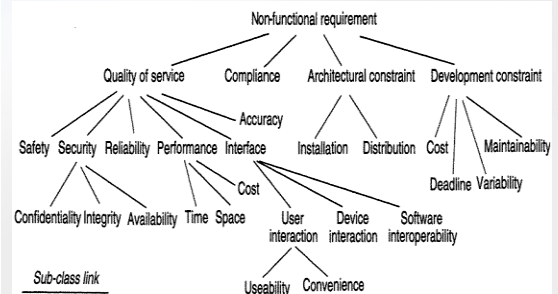
Catégories des besoins (1)

- 1- Besoins fonctionnels:** définissent les effets fonctionnels que le logiciel exige les avoir sur son environnement. Ils expriment les aspects de la dimension WHAT.
- Exemple: *le moteur de recherche bibliographique doit donner la liste de tous les livres sur un sujet donné.*
 - Résultats de ces besoins: opérations automatisées par le logiciel.
 - Caractérisent les fonctionnalités que le futur système doit les prendre en charge.

Catégories des besoins (2)

- 2- Besoins non fonctionnels:** définissent les contraintes sur la façon utilisée par le système futur pour satisfaire ses besoins fonctionnels, ou pour les développer.
- Exemples: *le format de requête et d'affichage des résultats doivent être compréhensibles par les étudiants non experts en informatique.*
 - Caractérisent une propriété ou une qualité désirée du système telle que sa performance, sa robustesse, sa convivialité, sa maintenabilité, etc.

Taxonomie des besoins non fonctionnels



Qualités des besoins(1)

- Complets:** l'ensemble de tous les besoins doit assurer la réalisation de tous les objectifs.
- Consistants:** les besoins doivent être compatibles les uns les autres (aucun besoin ne doit exclure un autre)
- Adéquats:** les besoins logiciel doivent être traduits adéquatement des besoins système. Décrits explicitement par les stakeholders.
- Non ambiguës:** exprimées d'une façon ne donnant pas confusion ou différentes interprétations.

Qualités des besoins(2)

- Mesurables:** les besoins doivent être formulés avec un degré de précision permettant:
 - Les analystes à mieux les évaluer
 - Les développeurs à mieux les tester
 - Les utilisateurs à mieux les localiser
- Pertinents:** Les besoins doivent contribuer à la réalisation des objectifs, mieux cerner le problème à résoudre.
- Réalisables:** vis-à-vis le budget, la planification et les contraintes technologiques.

Qualités des besoins(3)

- **Compréhensibles:** La formulation des besoins doit être compréhensible par les personnes qui les utilisent.
- **Bien structurés:** Le document des besoins doit être bien organisé dans la structure de ses éléments. La définition d'un terme doit précéder son utilisation.
- **Modifiables:** il doit être possible de réviser, corriger, étendre ou restreindre un besoin.
- **Traçables:** Le contexte dans lequel le document des besoins est créé doit être obtenu facilement. L'impact de modification, du suppression ou de création doit être fait facilement.

Défauts dans les documents des Besoins (1)

- RD: Requirements Document

Omission	Problem world feature not stated by any RD item – e.g. missing objective, requirement or assumption; unstated software response to some input.
Contradiction	RD items defining a problem world feature in an incompatible way.
Inadequacy	RD item not adequately stating a problem world feature.
Ambiguity	RD item allowing a problem world feature to be interpreted in different ways – e.g. ambiguous term or statement.
Unmeasurability	RD item stating a problem world feature in a way that cannot be precisely compared with alternative options, or cannot be tested or verified in machine solutions.
Noise	RD item yielding no information on any problem world feature.
Overspecification	RD item stating a feature not pertaining to the problem world but to the machine solution.

Défauts dans les documents des Besoins(2)

Unfeasibility	RD item that cannot be realistically implemented within the assigned budget, schedule or development platform.
Unintelligibility	RD item stated in an incomprehensible way for those who need to use it.
Poor structuring	RD items not organized according to any sensible and visible structuring rule.
Forward reference	RD item making use of problem world features that are not defined yet.
Remorse	RD item stating a problem world feature too late or incidentally.
Poor modifiability	RD items whose modification may need to be globally propagated throughout the RD.
Opacity	RD item whose rationale, authoring or dependencies are invisible.