

SI ET GENIE LOGICIEL

Conception de Systèmes d'Information

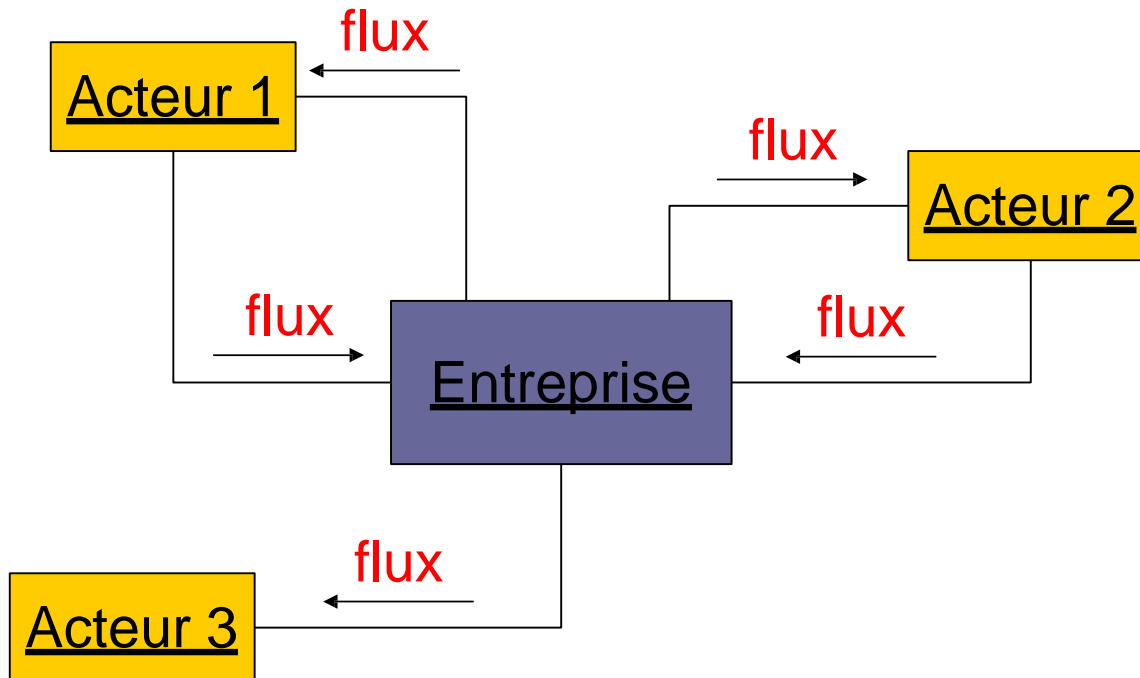
Ousmane SENGHOR

E-MAIL: o.senghor1919@gmail.com

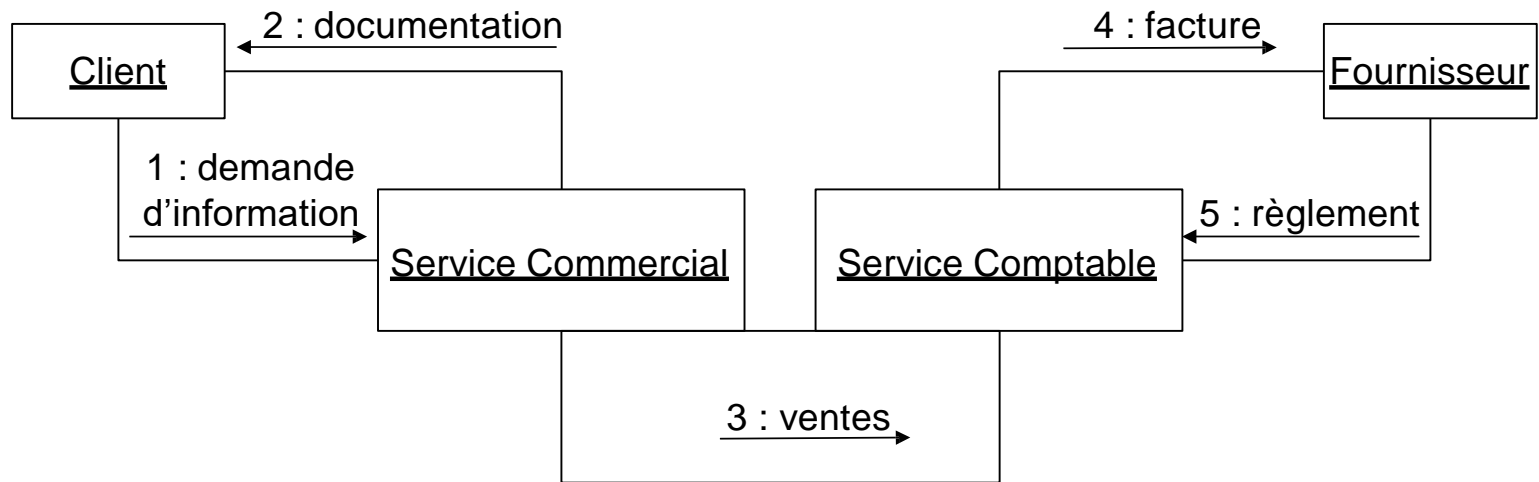
TEL: 77 459 41 71

- L'entreprise et son système d'information
- Problème actuels du génie logiciel
- Analyse vs conception
- Documentation et CASE
- Cycles de vie
- Modèle
- Notions sur les approches orientées objet
- Introduction à UML

L'entreprise et son environnement



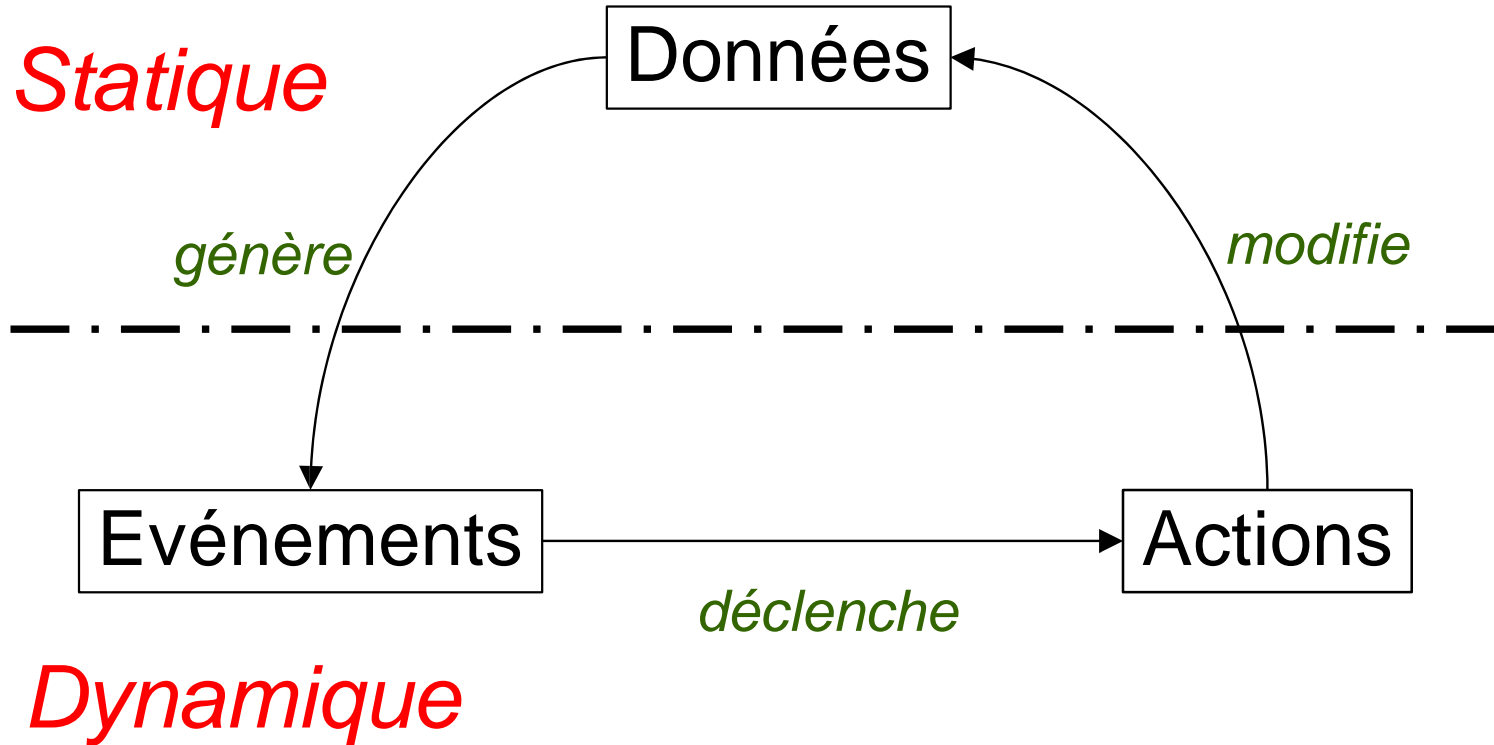
Les domaines de l'entreprise



Degrés d'invariance dans l'entreprise

- Ce qui est stable
 - processus liant l'entreprise à ses acteurs externes
 - données
- Ce qui est moins stable
 - traitements
- Ce qui est peu stable
 - techniques
 - nature de la demande
 - besoins en statistiques
 - organisation de l'entreprise

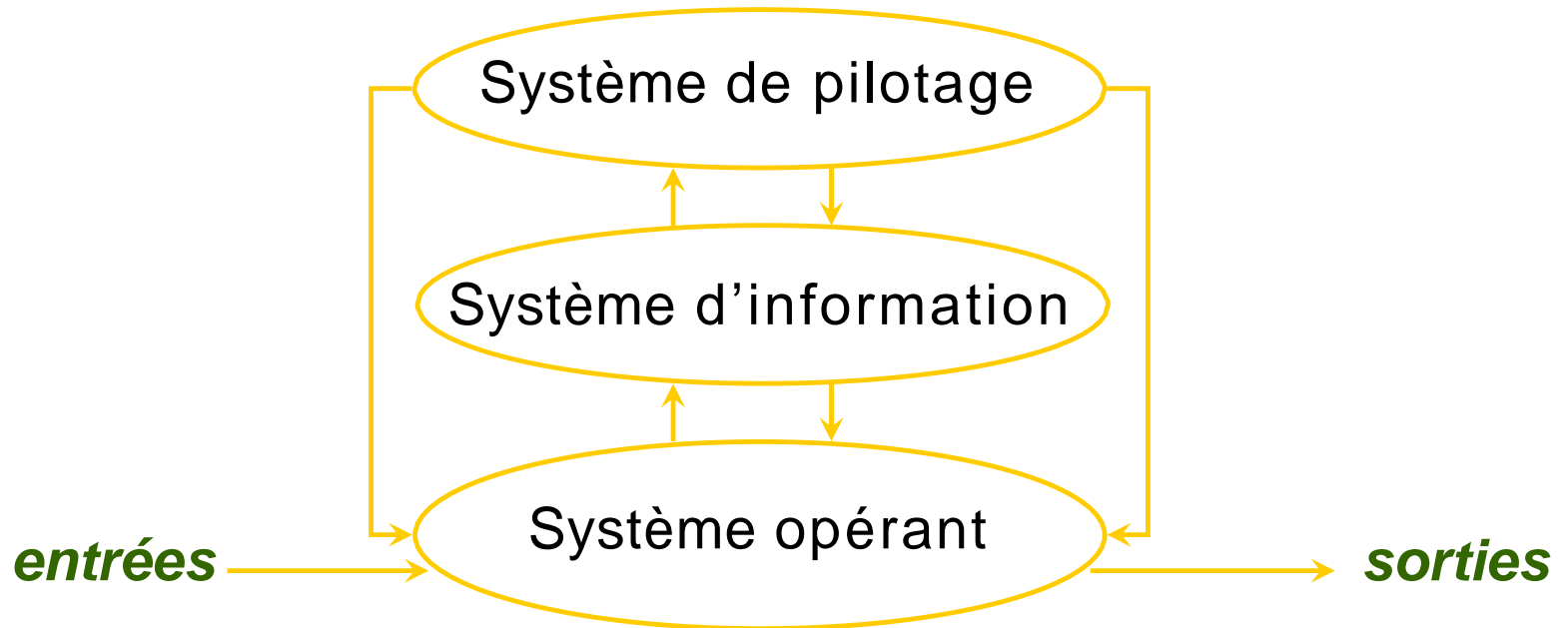
Statique et dynamique du système d'information



A quoi sert l'information ?

- Support pour l'action
- Conserve une trace des activités
- Apporte une aide à la décision
- Technologie de l'information et de la communication
 - Mémorisation
 - traitement automatique
 - Diffusion

Le système d'information dans l'entreprise



Le schéma directeur

- Définit les orientations stratégiques de l'entreprise en matière de système d'information :
 - Politique d'investissement matériel et logiciel
 - Choix d'organisation des systèmes d'information (centralisation/répartition)
 - Rôle des différents acteurs de la politique informatique (utilisateurs, techniciens, exploitants)

- L'ensemble des actions ayant trait au traitement de l'information dans l'entreprise devra être cohérent avec le schéma directeur

- L'entreprise et son système d'information
- Problème actuels du génie logiciel
- Analyse vs conception
- Documentation et CASE
- Cycles de vie
- Modèle
- Notions sur les approches orientées objet
- Introduction à UML

Problèmes actuels du génie logiciel

- Taille et complexité du logiciel
 - Complexité fonctionnelle
 - Mutations technologiques perpétuelles
 - Complexité des architectures
 - Solutions :
 - Distinguer analyse et réalisation
 - Décomposer le système
 - Utiliser une approche de haut niveau

Problèmes actuels du génie logiciel

- Taille croissante des équipes
 - Compétences de + en + variées et pointues
 - Applications stratégiques orientées métier
 - Délais de + en + courts
 - Solutions :
 - Technologie unifiant le vocabulaire
 - Méthode, démarche de travail

Problèmes actuels du génie logiciel

- Évolution rapide des applications
 - Besoins du client
 - Activité du client
 - Environnement technique
 - Solution :
 - Cycle de vie itératif et incrémental

Problèmes actuels du génie logiciel

- Spécifications peu précises
 - Imprécision, incomplétude
 - Interface difficile entre domaine métier et informatique
 - Solution :
 - Utilisation de modèles, notamment graphiques

Exercice : l'argent de la caisse

Le dernier client venait de quitter le magasin. L'un des propriétaires ramassait le contenu d'une caisse enregistreuse quand un homme rentra. L'inconnu alla droit vers le gérant et lui demanda de l'argent.

La lumière s'éteignit brusquement. Quand elle revint, l'inconnu avait disparu. Toutes les caisses enregistreuses étaient vides.

L'inspecteur Lapreuve arriva immédiatement sur les lieux.

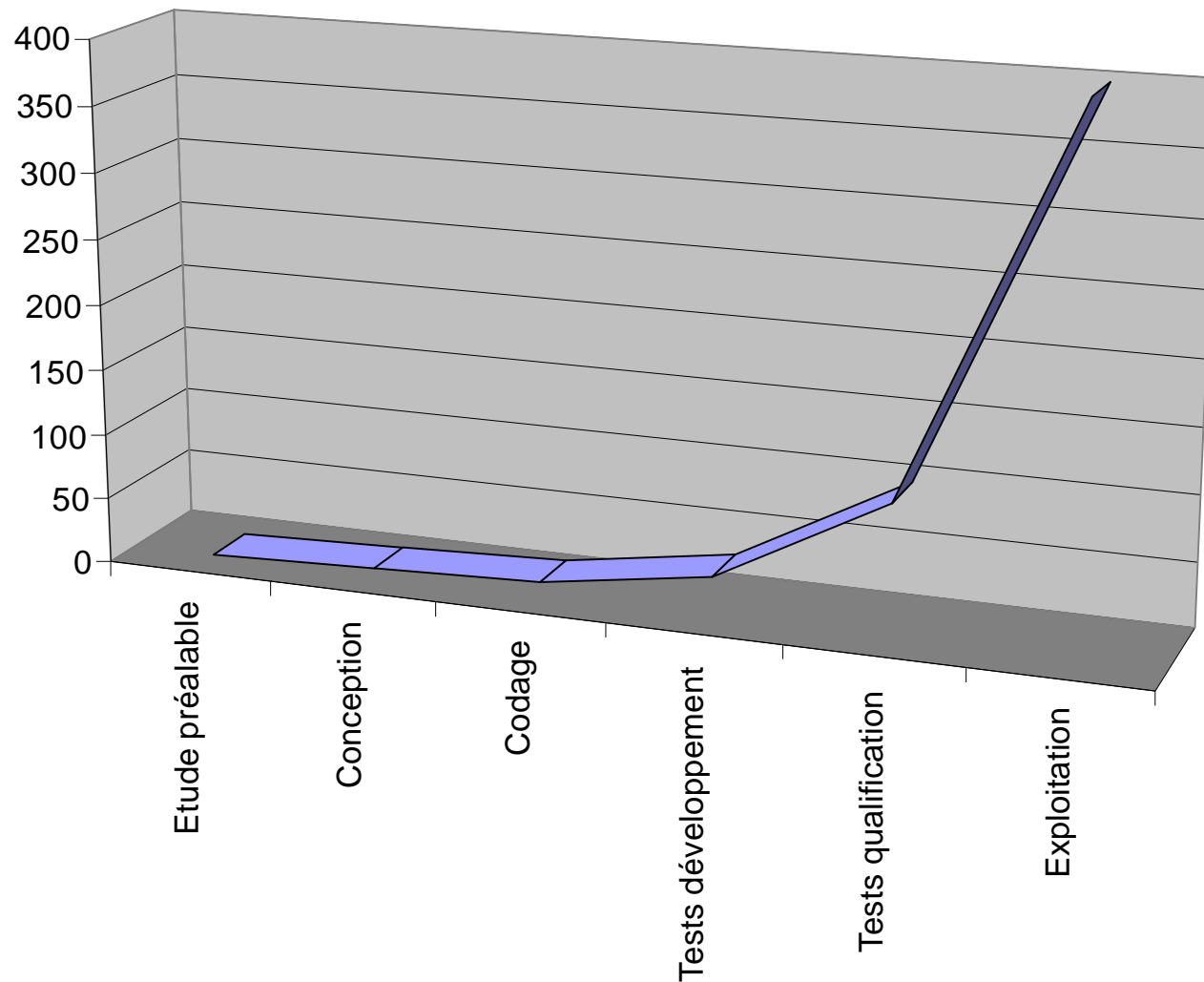
Cocher les cases inappropriées

	oui	non	?
1. L'inconnu s'adressa au gérant			
2. Le voleur ne demanda pas d'argent			
3. Il n'y a qu'un propriétaire			
4. Le voleur a coupé l'électricité			
5. L'histoire ne précise pas combien d'argent a disparu			
6. Seules 2 personnes étaient présentes quand l'homme entra			
7. Le voleur voulait de l'argent			
8. Le gérant ramassait le contenu des caisses enregistreuses			
9. Le propriétaire a reconnu l'inconnu			
10. L'inspecteur Lapreuve recherchait l'inconnu			

Défauts du langage humain

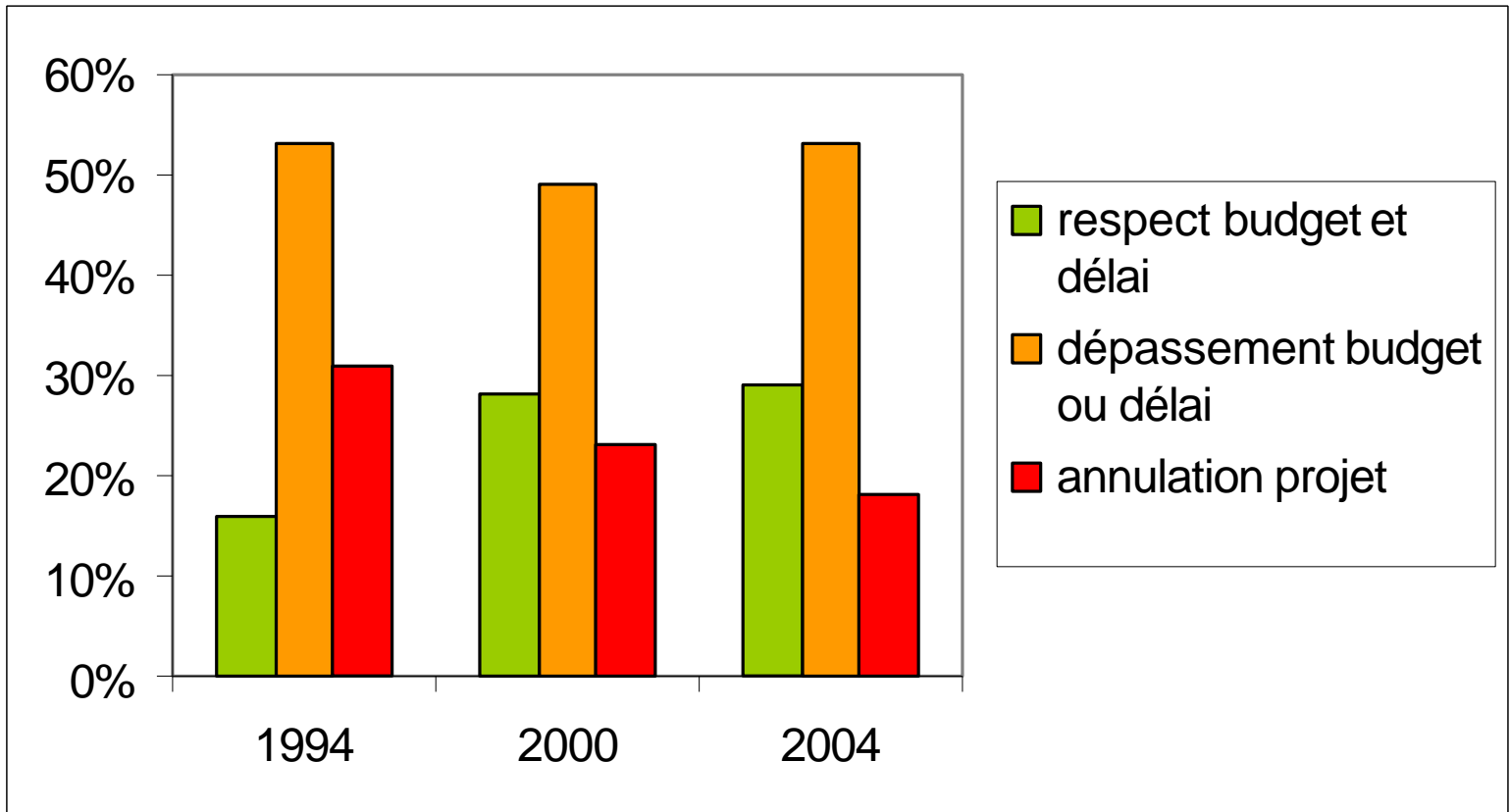
- Bruit
- Silence
- Sur-spécification
- Contradiction
- Ambiguïté
- Référence en avant
- Vœu pieu

Impact d'une erreur de spécification



- L'entreprise et son système d'information
- Problème actuels du génie logiciel
- Analyse vs conception
- Documentation et CASE
- Cycles de vie
- Modèle
- Notions sur les approches orientées objet
- Introduction à UML

Une amélioration lente



Source : Standish group, 2006

Nécessité de l'analyse et de la conception

□ Exemple de problème

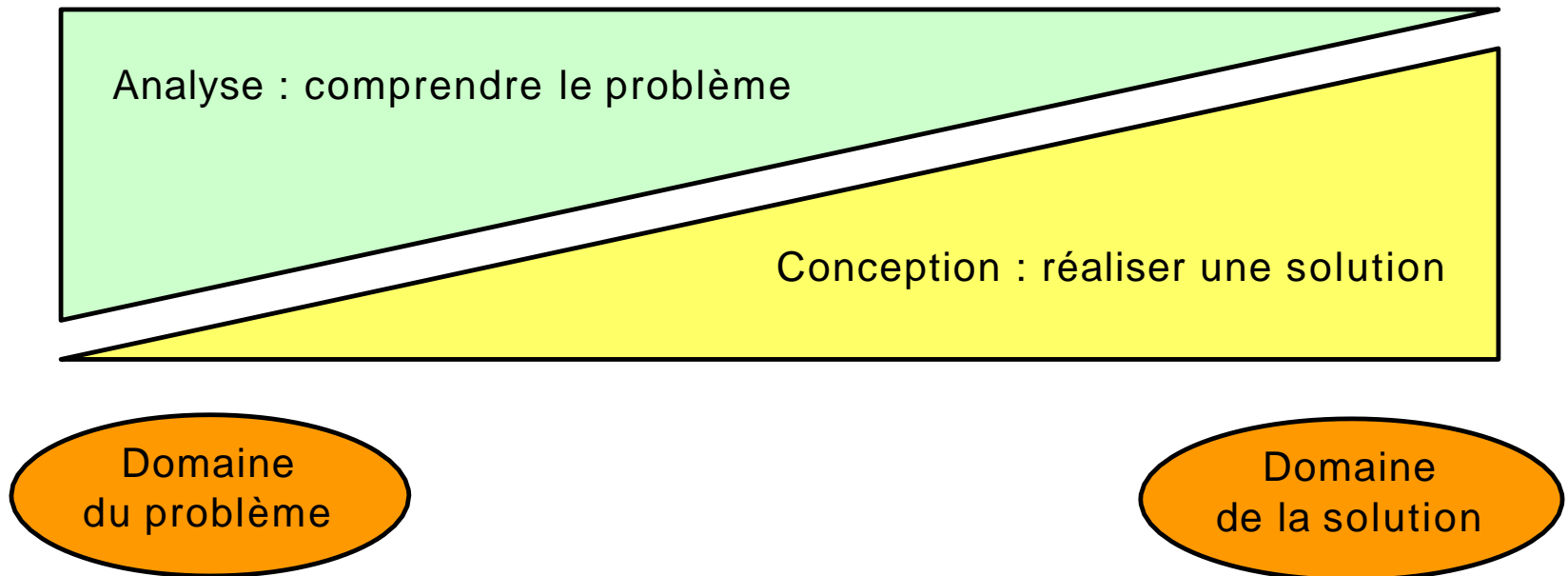


□ Proposer quelques solutions

Poser les bonnes questions

- QUEL est le problème ?
- POURQUOI le problème existe-t-il ?
- QUI est impliqué ?
- OU se situe le problème ?
- QUAND faut-il mettre en œuvre la solution ?
- QUELLE technologie est implicitement pressentie ?

Analyse vs conception



□ La conception est souvent un compromis

Analyse vs conception

- L'analyse prend en compte l'environnement du monde réel
 - QUOI
 - Conceptuel
 - Peut couvrir des besoins de l'utilisateur hors logiciel

- La conception consiste à trouver une solution techniquement possible
 - COMMENT
 - Organisation, implémentation physique

- Comprendre le problème
 - Parfois, un cahier des charges textuel est rédigé par l'utilisateur
 - Poser des questions pour identifier les vrais besoins
 - Supprimer les ambiguïtés et incohérences, rendre précis, faire émerger les « non-dits »
 - Parler le langage de l'utilisateur (le client)
- Réaliser des modèles (abstractions)
 - L'ensemble des modèles constitue le modèle d'analyse
 - C'est une représentation + ou - formelle de l'information
- Le modèle d'analyse est contractuel
 - C'est un support de communication entre l'utilisateur et le concepteur
- En amont, il faut cerner les besoins par une étude préalable du domaine

□ Résoudre le problème

■ Connaître

- Les technologies
- Les architectures appropriées
- Les bonnes pratiques

■ Comparer les approches possibles (trade-off)

■ Appliquer des solutions standards quand c'est possible (design patterns, architectures de domaines (frameworks))

■ Parler le langage de l'utilisateur et de l'analyste

□ Réaliser le modèle de conception

■ Adapter le modèle d'analyse

■ Allouer les composants du modèle d'analyse à des composants du modèle de conception

- L'entreprise et son système d'information
- Problème actuels du génie logiciel
- Analyse vs conception
- Documentation et CASE
- Cycles de vie
- Modèle
- Notions sur les approches orientées objet
- Introduction à UML

Importance de la documentation

- La réalisation de la documentation est perçue comme une contrainte
 - Elle sera faite plus tard (= jamais)
 - Elle n'est pas mise à jour
- Les spécifications et la conception doivent être documentées
 - Approbation des clients
 - Compréhension des développeurs
- Une bonne stratégie de tests peut être développée à partir des spécifications
- La documentation est indispensable pour la maintenance

□ Computer Aided Software Environment

- Éditeur de diagrammes
- Dictionnaire de données (repository)
- Contrôle d'accès
- Vérifications automatiques de la cohérence, complexité...
- S'appuie sur une méthode
- Génération automatique de documentation
- Génération automatique de code

□ Exemples

- Magic Draw UML 11.5 (No Magic, Inc.)
- Objecteering (Softeam)
- Rose (Rational)
- Mega (IBM)
- Class Builder (*freeware*)
- Outils gratuits de démonstration : UML Jude, visual paradigm (community)

- L'entreprise et son système d'information
- Problème actuels du génie logiciel
- Analyse vs conception
- Documentation et CASE
- Cycles de vie
- Modèle
- Notions sur les approches orientées objet
- Introduction à UML

Démarche pour maîtriser et bâtir une application

- Déterminer les buts
- Définir le processus (cycle de vie)
- Utiliser une méthode
- S'appuyer sur un formalisme

Cycle de vie en cascade

□ Spécification du logiciel

Ensemble des activités consistant à définir de manière précise, complète et cohérente ce dont l'utilisateur a besoin

□ Conception préliminaire

Ensemble des activités conduisant à l'élaboration de l'architecture du logiciel

□ Conception détaillée

Ensemble des activités consistant à détailler les résultats de la conception préliminaire, tant sur le plan algorithmique que sur celui de la structure des données, jusqu'à un niveau suffisant pour permettre le codage

□ Codage

Activité permettant de traduire le résultat de la conception détaillée en un programme à l'aide d'un langage de programmation donné

Cycle de vie en cascade

□ Tests unitaires

Activité ayant pour but de vérifier pour chaque composant du logiciel pris isolément que :

- Tous les chemins logiques sont parcourus au moins une fois
- La plage de validité des données d'entrée et de sortie a été explorée
- Les résultats sont conformes au dossier de conception détaillée

□ Intégration

Activité consistant à assembler progressivement les composants du logiciel identifiés lors de la phase de conception préliminaire et contrôlés lors des tests unitaires

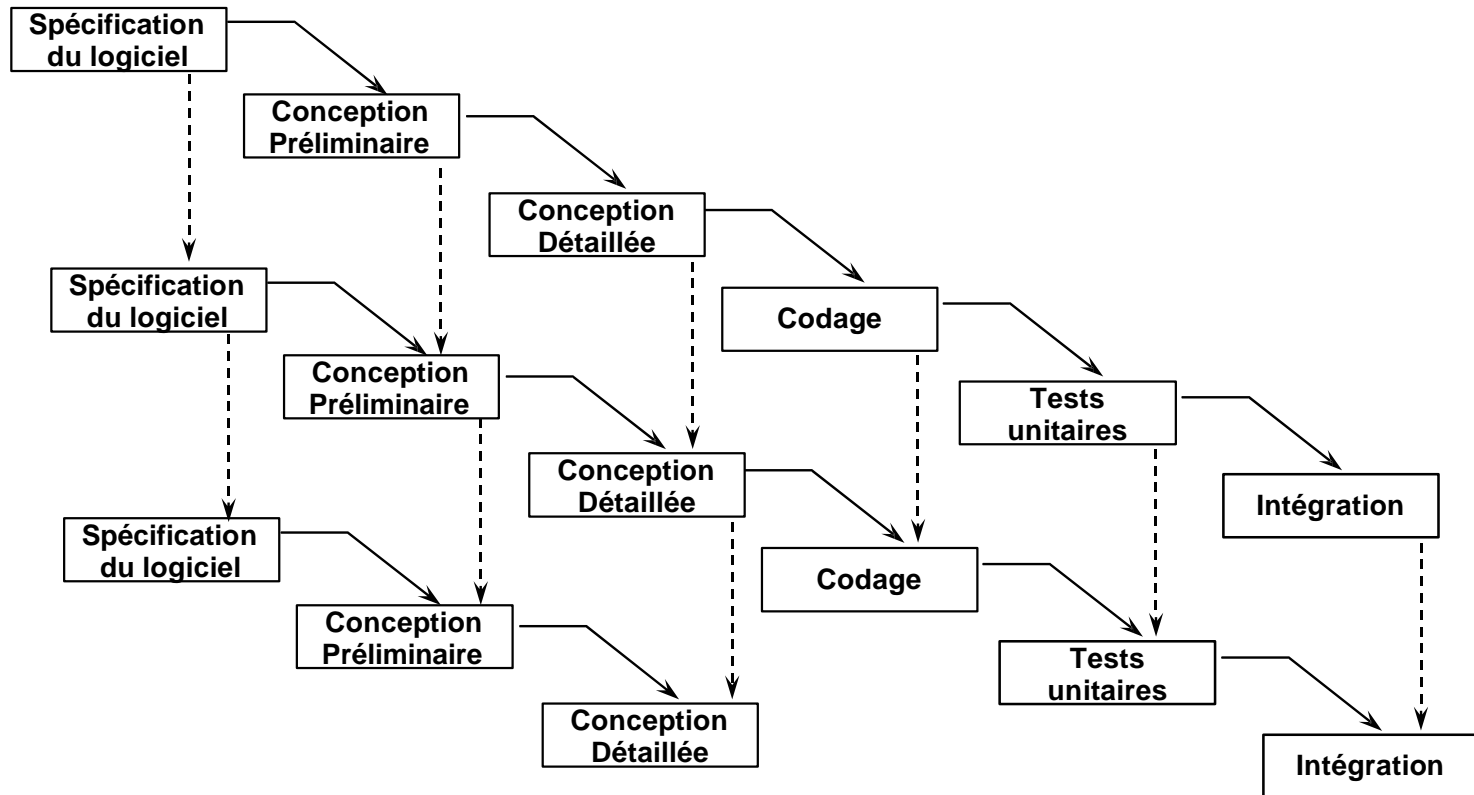
□ Validation

Activité conduisant à s'assurer, essentiellement au moyen de tests, qu'un logiciel est conforme au dossier de spécifications du logiciel

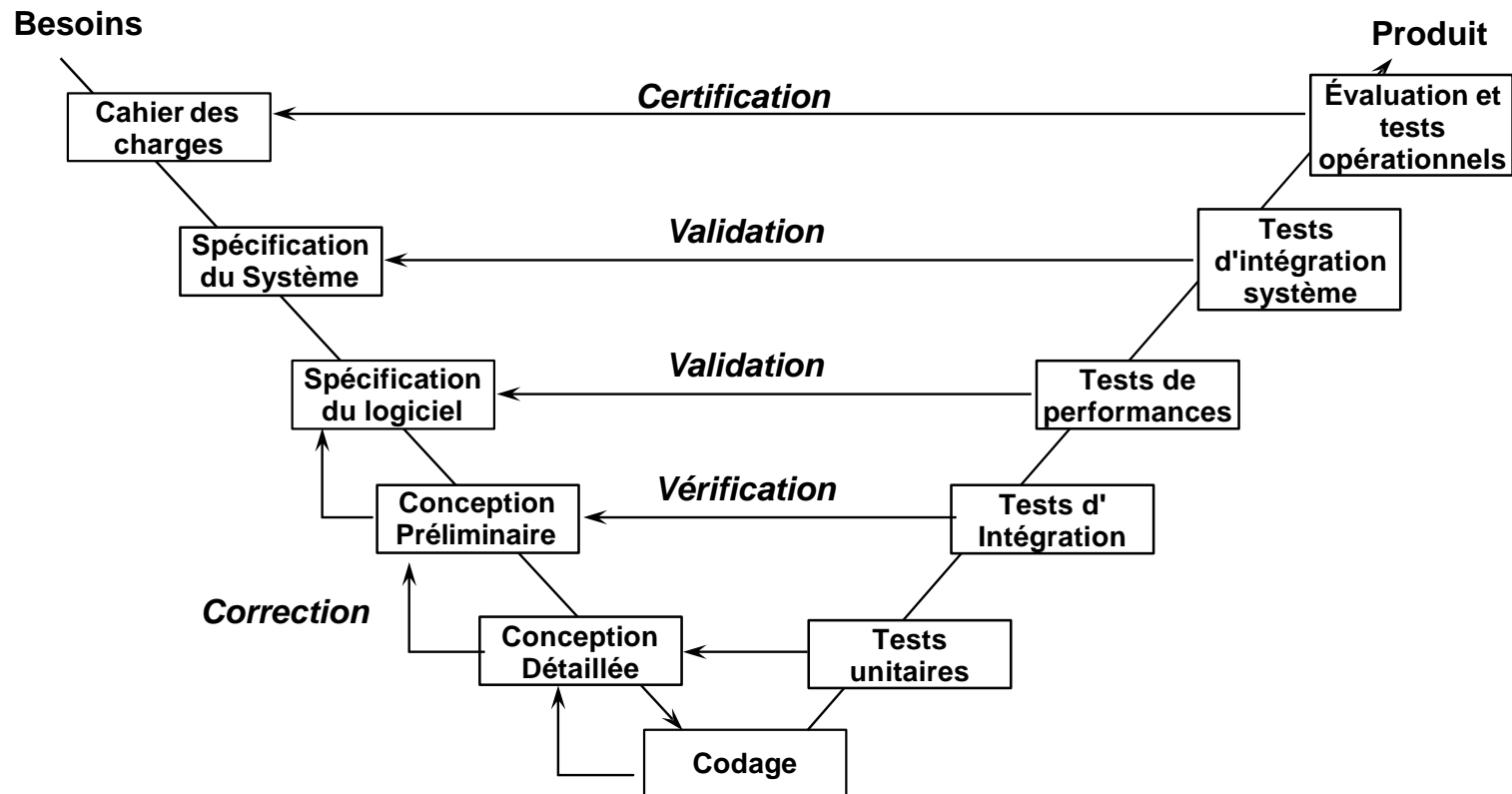
□ Exploitation

- Exploitation du logiciel
 - Ensemble des activités liées à la mise en œuvre opérationnelle d'un logiciel dans un environnement déterminé
- Utilisation du logiciel
 - Ensemble des activités liées au besoin pour lequel le logiciel a été développé
- Maintenance du logiciel
 - Ensemble des activités liées à la détection et la correction des défauts résiduels
- Adaptation du logiciel
 - Ensemble des activités liées aux évolutions du dossier de spécification du logiciel

Cycle Incrémental



Cycle en V



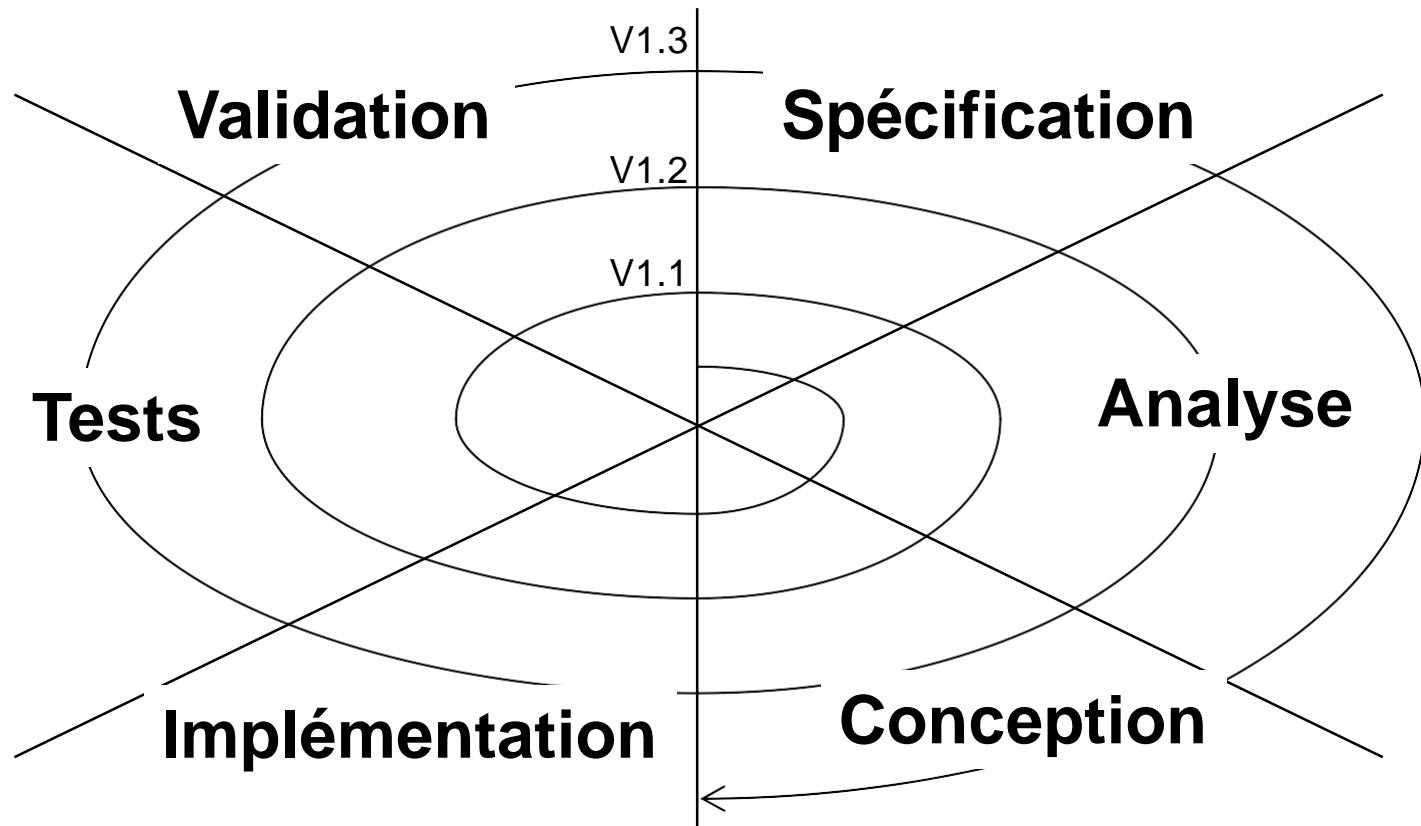
Avantages et limites Cycle en V

- A fait ses preuves sur de gros projets
- Le plus utilisé, le mieux compris
- Nombreuses opportunités de retour d'information
- N'est qu'un modèle dont la réalité s'écarte souvent
- Ne prend pas en compte l'aspect économique

Le cycle de vie objet

- 3 caractéristiques fondamentales :
 - Traçabilité entre les étapes
 - Correspondance aisée entre les éléments définis dans 2 phases successives
 - Une difficulté : passer des besoins (fonctionnels) à l'analyse (objets)
 - Caractère itératif
 - Caractère incrémental

Cycle en spirale



Avantages et limites du cycle en spirale

- Permet une grande flexibilité
- Très utile si les exigences initiales sont peu claires ou fortement évolutives
- Bien adapté aux technologies nouvelles
- Se prête difficilement aux plans à long terme
- Ne prend pas en compte l'aspect économique

Validation des étapes par des revues (recettes)

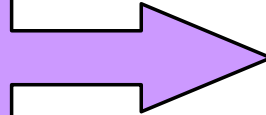
- Les revues doivent être courtes et bien ciblées
 - 1h maxi
- Ne pas entrer dans des discussions improductives
 - Règle des 3mn
 - Si le débat sur un point nécessite plus de temps, noter le problème
 - Réponse lors de la revue suivante
- Livrables
 - Fiches de recette
 - Modifications à apporter
 - Liste de questions

Le but est de
trouver des erreurs
ou omissions
Pas de les corriger



- L'entreprise et son système d'information
- Problème actuels du génie logiciel
- Analyse vs conception
- Documentation et CASE
- Cycles de vie
- Modèle
- Notions sur les approches orientées objet
- Introduction à UML

Services attendus
par l'utilisateur



le QUOI
et pas
le COMMENT

Cohérent et complet

Systèmes
complexes



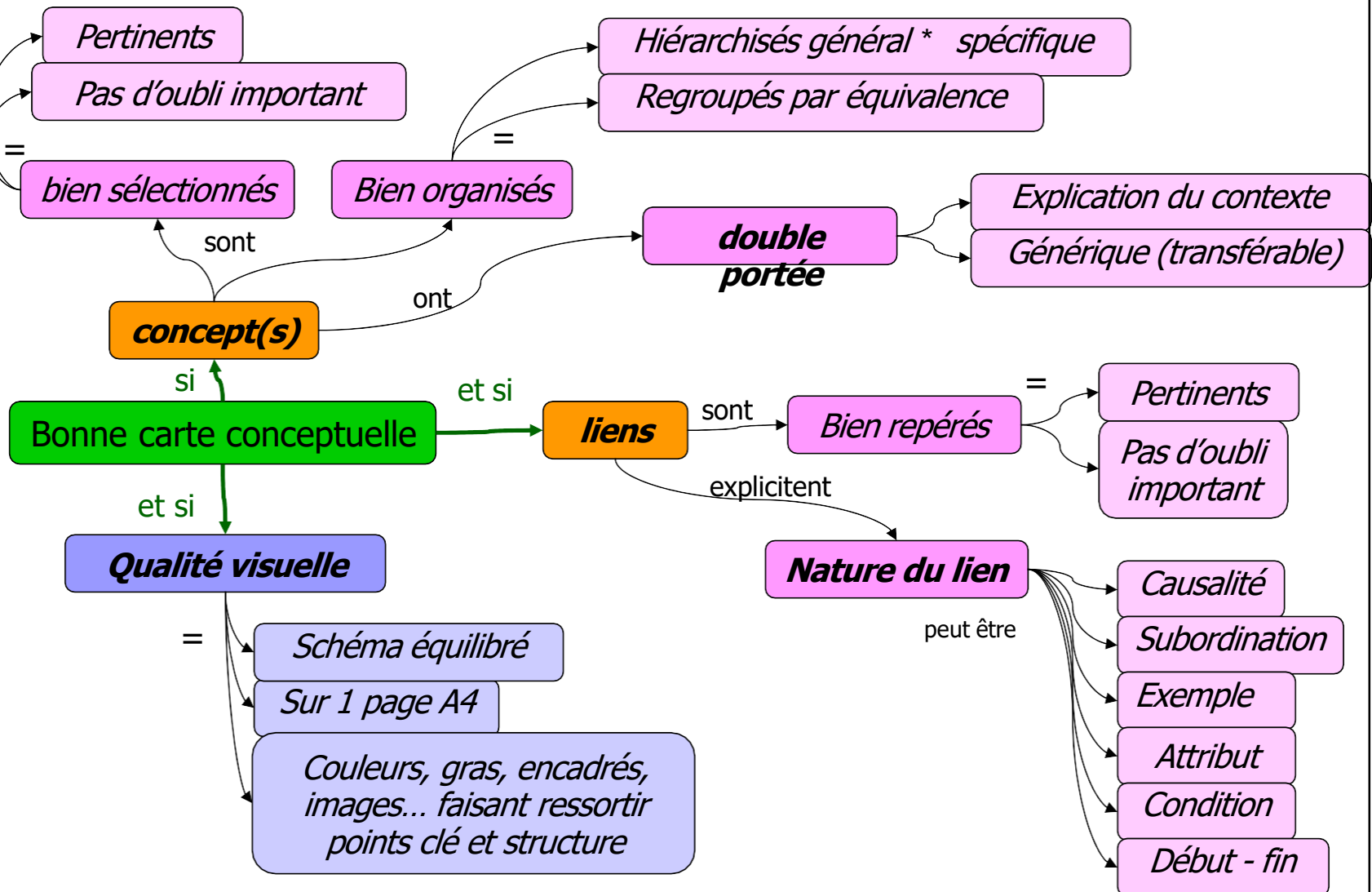
**Modèles
graphiques**

(cours C Verzat)



Qu'est-ce qu'une bonne carte conceptuelle ?

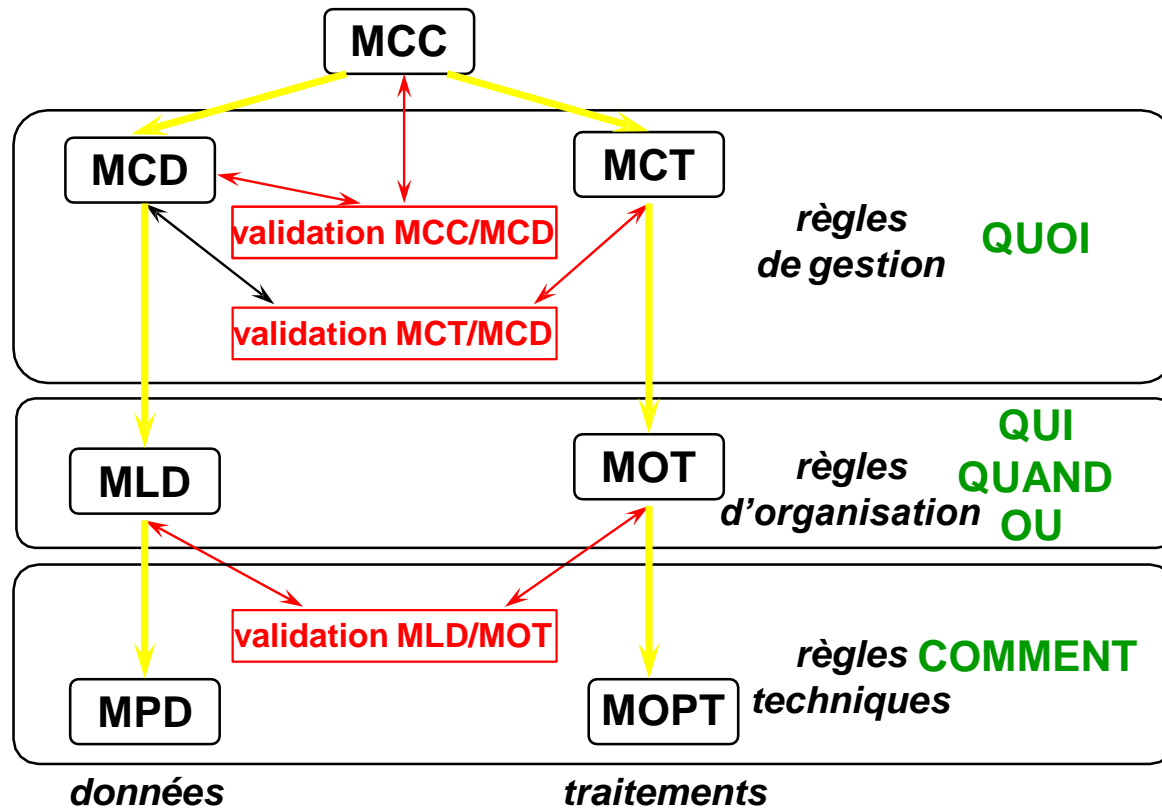
(cours C Verzat)



Trois approches de modélisation

- Modélisation des données
 - Quoi
- Modélisation des comportements
 - Quand, comment
- Modélisation objet
 - Quoi, quand, comment

Cycle d'abstraction Merise



- L'entreprise et son système d'information
- Problème actuels du génie logiciel
- Analyse vs conception
- Documentation et CASE
- Cycles de vie
- Modèle
- Notions sur les approches orientées objet
- Introduction à UML

Pourquoi l'approche objet ?

- Stabilité de la modélisation par rapport aux entités du monde réel
- Construction itérative facilitée par le faible couplage entre composants
- Possibilité de réutiliser des éléments
- Simplicité du modèle :
 - 5 concepts : objets, messages, classes, généralisation, polymorphisme

Pourquoi l'approche objet ?

- Stabilité de la modélisation par rapport aux entités du monde réel
- Construction itérative facilitée par le faible couplage entre composants
- Possibilité de réutiliser des éléments
- Simplicité du modèle :
 - 5 concepts :
 - Objets
 - Classes
 - Messages
 - Généralisation
 - Polymorphisme

□ Approche objet : encapsulation

Objet = état + comportement + identité

□ Exemple :

Un objet Personne



Informations

nom = « BOND »

âge = 35 ans

société = « MI6 »

Comportements

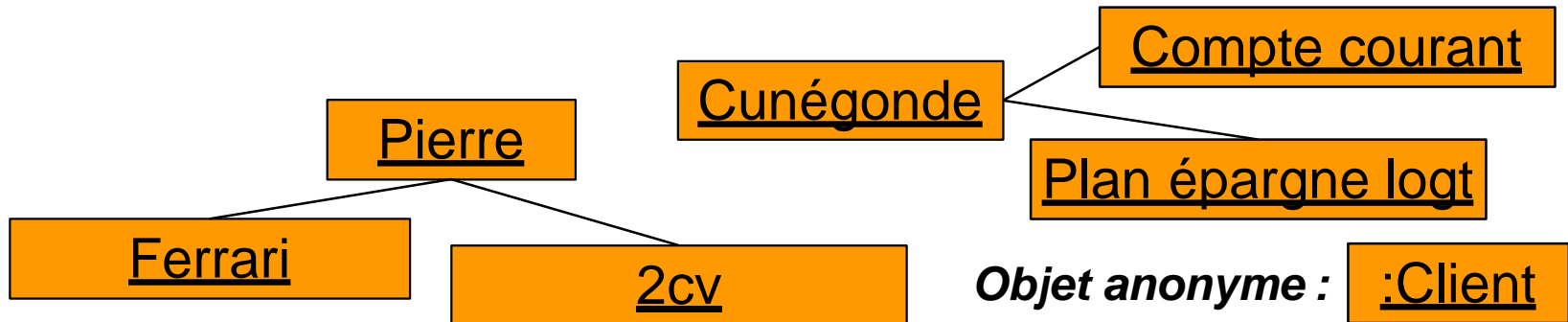
nomEnMajuscule

changerAge

changerSociété

Caractéristiques d'un objet

- Objet : peut être concret ou abstrait



- Intérêt de la relation d'encapsulation :
 - Forte cohésion interne
 - Faible couplage avec l'extérieur
- Chaque objet possède une identité unique (non explicitée sur le modèle)
- État : regroupe les valeurs de tous ses attributs à un instant donné

- Classe (d'objets) : abstraction d'objets ayant :
 - Des propriétés similaires
 - Un comportement commun
 - Des relations identiques avec les autres objets
 - Une sémantique commune

- Tout objet connaît sa classe
 - Propriété implicite d'un objet
 - Pb : changement de classe (transmutation)

- Classe : abstraction du problème, généralisation

Exemple de classe

Une instance de Personne

nom = « BOND »
âge = 35 ans
société = « MI6 »

instanciation

Classe Personne

Informations

nom
âge
société

Comportements

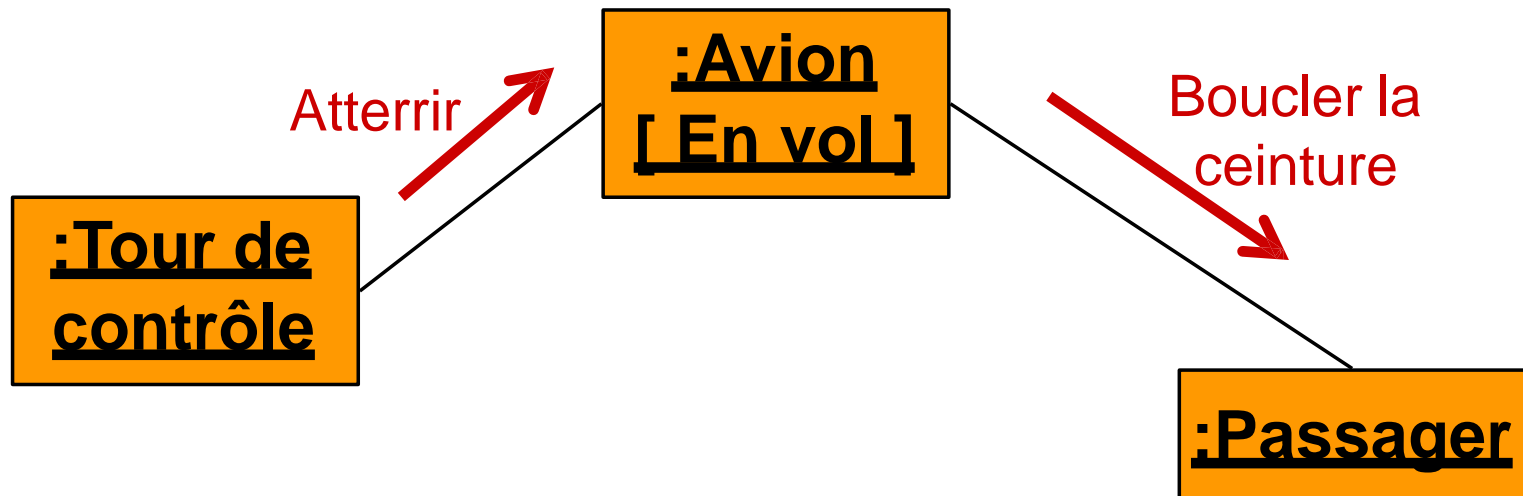
nomEnMajuscule
changerAge
changerSociété

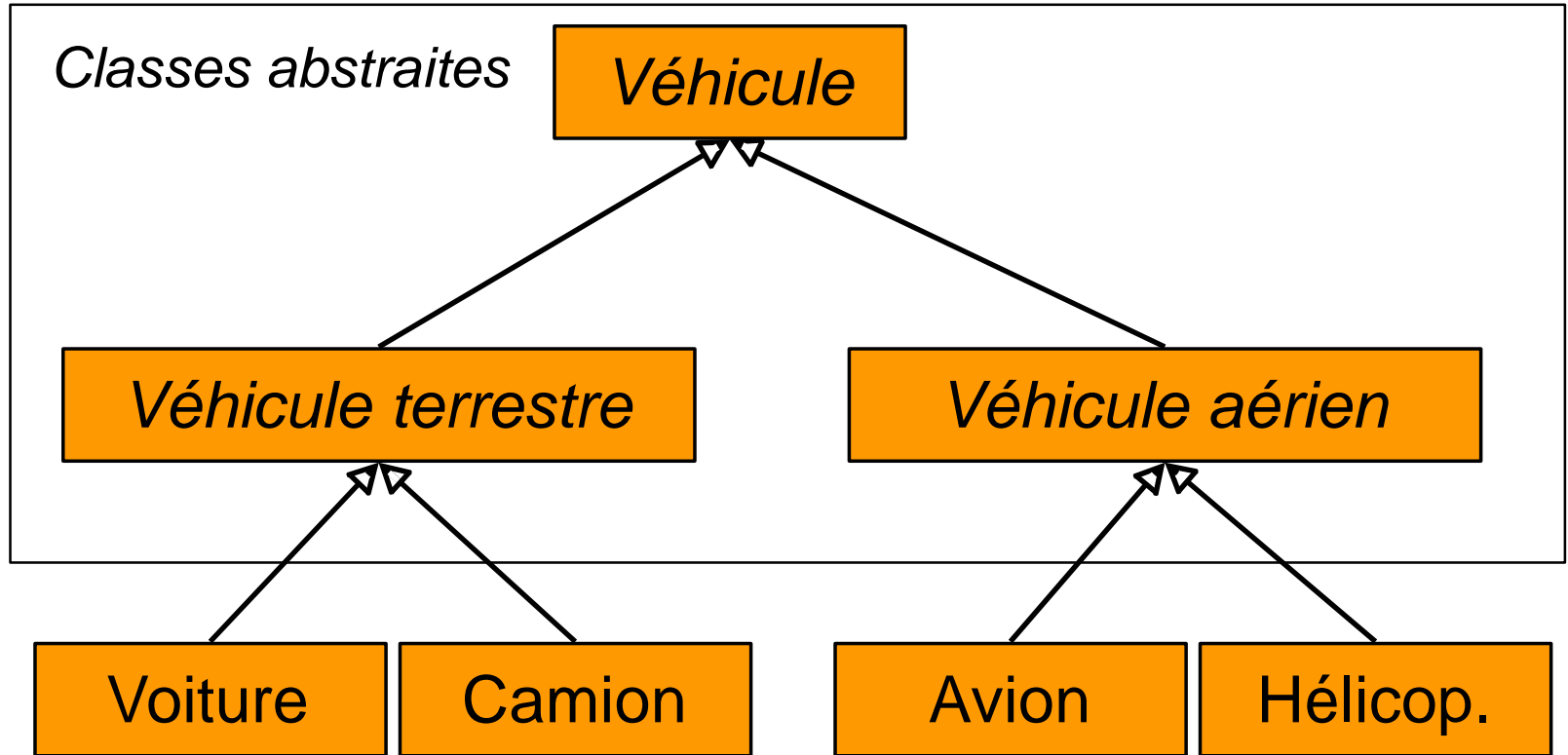
Une instance de Personne

nom = « MATA ARI »
âge = 32 ans
société = « KGB »

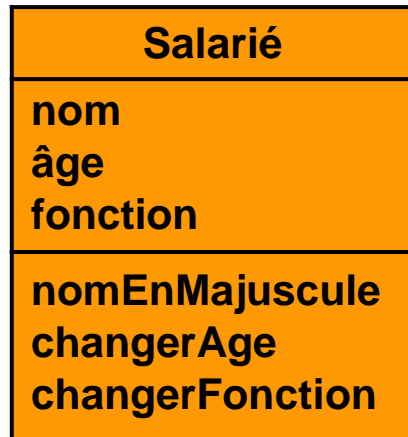
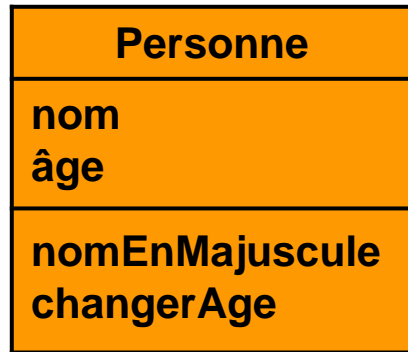
instanciation

- Message : stimulation externe d'un objet par un autre
- Les opérations (actions et réactions) sont déclenchées suite à l'envoi d'un message par un autre objet

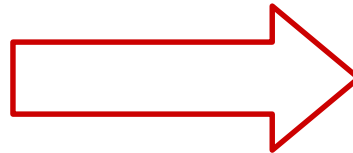




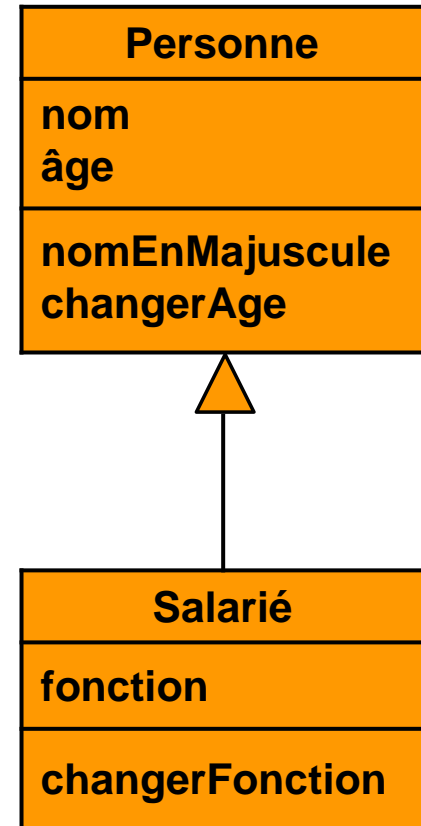
Généralisation - spécialisation



généralisation



spécialisation



Polymorphisme

- Un nom d'objet peut désigner des instances de classes différentes issues d'une même arborescence
- L'objet peut réagir différemment à des opérations communes

