

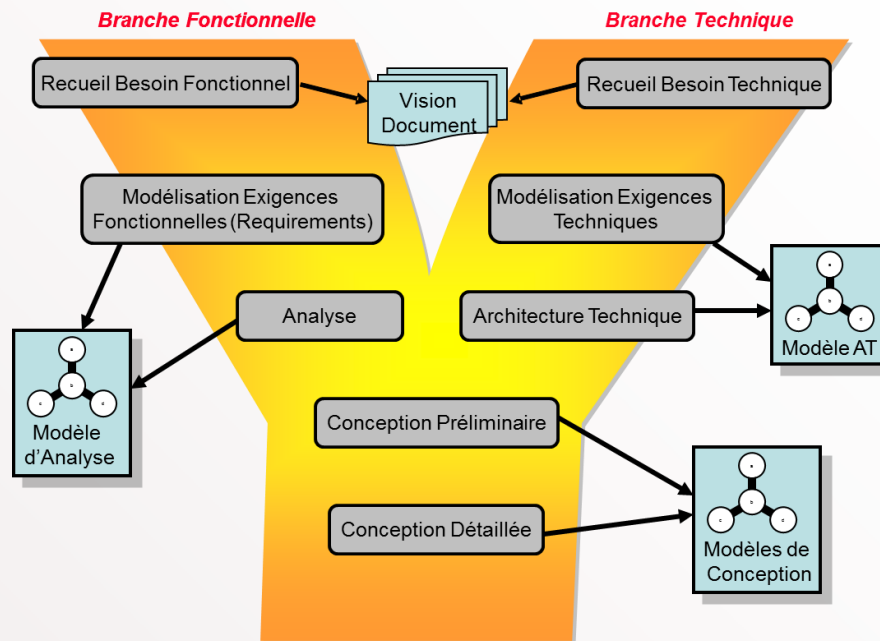


Conduite de Projets

Ingénieurs 2^o et 3^o année

IV b – Modélisation du besoin

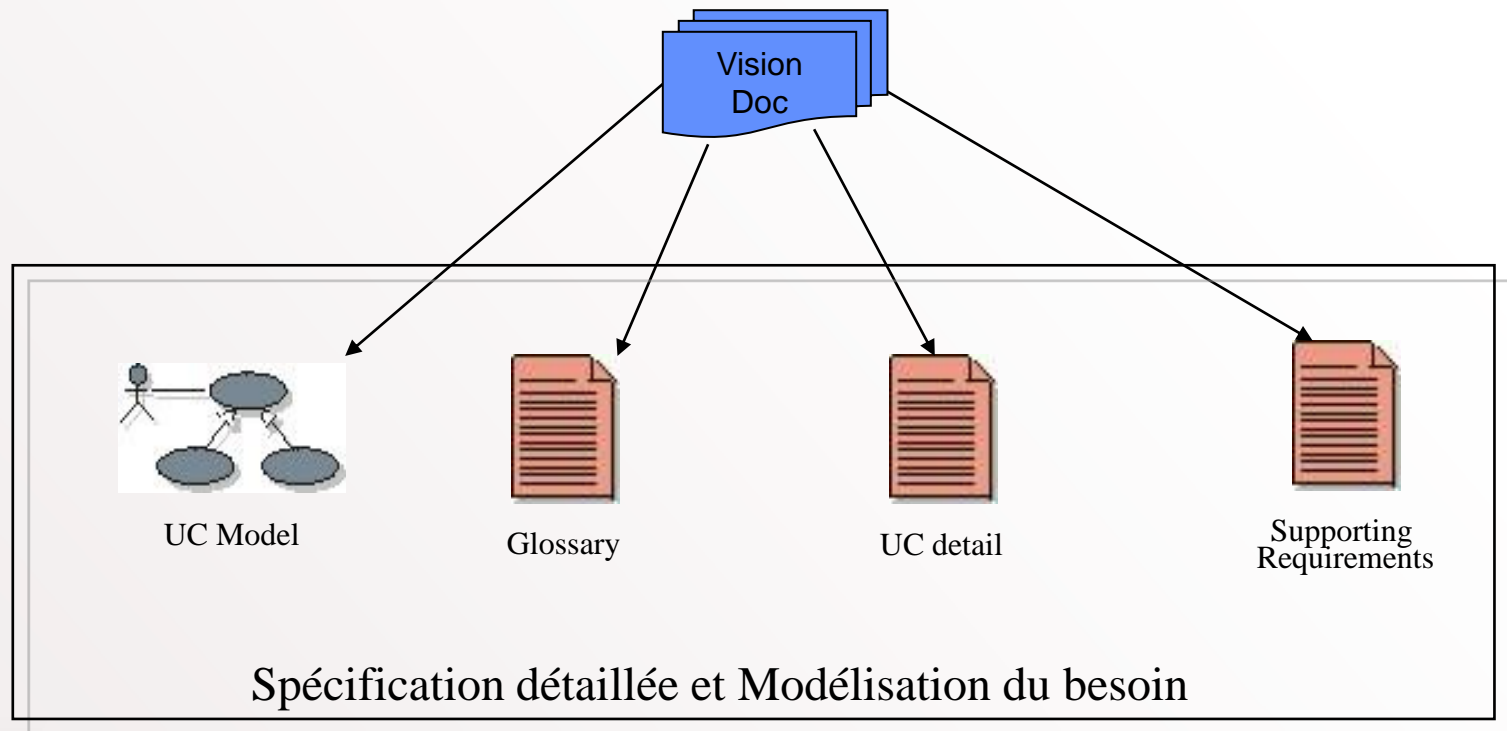
1 – Modélisation du besoin



Introduction

❑ Vision Document: exhaustif mais peu détaillé.

➔ Affiner et détailler pour préparer l'analyse et la conception (approche UP).



UP vs Agile

❑ Expression du besoin en UP

- UC Model fondamental – Nombreux modèles UML
- Spécifications robustes, enrichies itérativement
- Approche assez formalisée

❑ ...en Agile (extrême)

- Analyses et hypothèses initiales souvent minimalistes. Parfois un Vision Document
- Spécifications minimalistes ou « Modélisation Agile »
- Très peu de livrables (choix libre de l'équipe)

❑ Modélisation UP: deux approches complémentaires

- **Modèle de UC**: description des processus fonctionnels (interactions entre le S.u.d. et ses acteurs).

C'est un modèle dynamique.

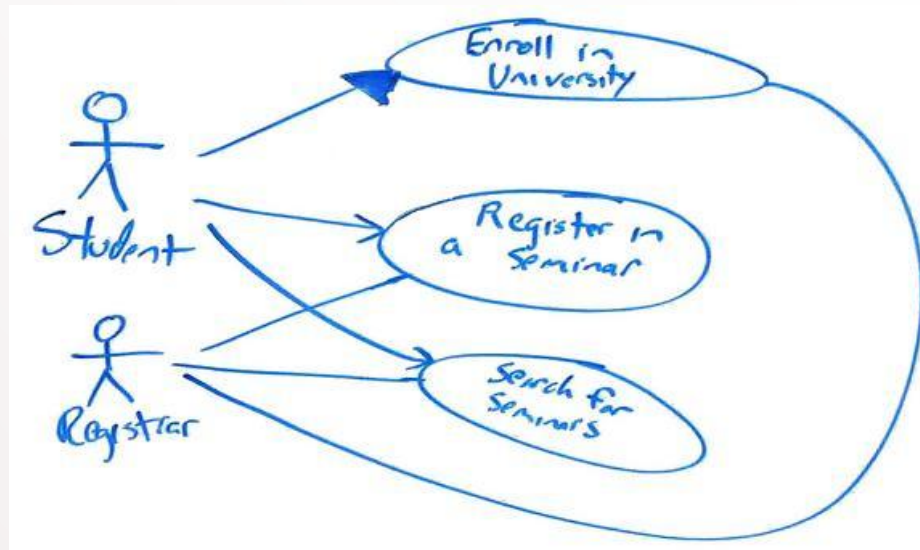
➔ Débouche sur les UC details (scénarios) et des Supporting Requirements.

- **Modèle du Domaine**: description des grands concepts d'information manipulés par le métier dans le cadre de l'utilisation du S.u.d.

C'est un modèle statique.

➔ Débouche sur la rédaction d'un Glossaire puis diagramme de Classes métier.

2 – UC Model



Elaborer un UC model

- ❑ Identifier les UC et les acteurs
- ❑ Décrire brièvement les UC et acteurs:
 - très succinct à ce stade, quelques lignes; il ne s'agit que d'avoir une vision à haut niveau, le « brief » du UC.
- ❑ Etablir les relations entre UC et entre acteurs: extends, includes, généralisations,...
- ❑ Structurer les UC en packages
- ❑ Décrire les packages
- ❑ Tracer le diagramme de UC

Use-Case

“

Service rendu par le
système à son utilisateur
(dans le cadre de son métier)

ou

L'intention de l'utilisateur
quand il manipule le
système.

“

Use-Case

*Il est toujours exprimé
sous la forme d'un verbe à
l'infinitif traduisant cette
intention.*

Identification des UC

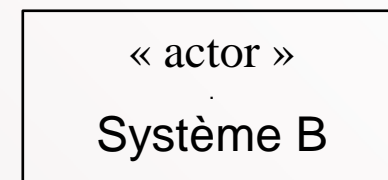
- ❑ Chaque UC représente une façon d'utiliser le système = définir ce que doit faire le système.
- ❑ Le UC doit apporter une valeur ajoutée significative à l'utilisateur et à l'entreprise.
- ❑ L'identification des UC requiert donc de connaître en détail les besoins des utilisateurs.
 - Pour cela il faut comprendre le métier, interroger les utilisateurs, ...

Granularité des UC

- ❑ Il est difficile de déterminer la bonne granularité d'un UC:
 - Trop gros: modélisation lourde, difficile de planifier l'implémentation sur un cycle d'itérations courtes.
 - Trop petit: séquence d'actions atomiques → dizaines de UC par système sans vue d'ensemble.
- ❑ Le nombre théorique idéal : 15-20 UC significatifs par système ou sous-système.

Identification et Description des Acteurs

- ❑ Utilisateurs du Vision Document = base à l'identifications des acteurs.
- ❑ Un même utilisateur peut avoir plusieurs rôles par rapport au système → plusieurs acteurs.
- ❑ Les acteurs peuvent être modélisés avec des relations de généralisation/spécialisation.
- ❑ On distingue les acteurs humains des acteurs système (systèmes externes au S.u.d. mais interfacés avec celui-ci).

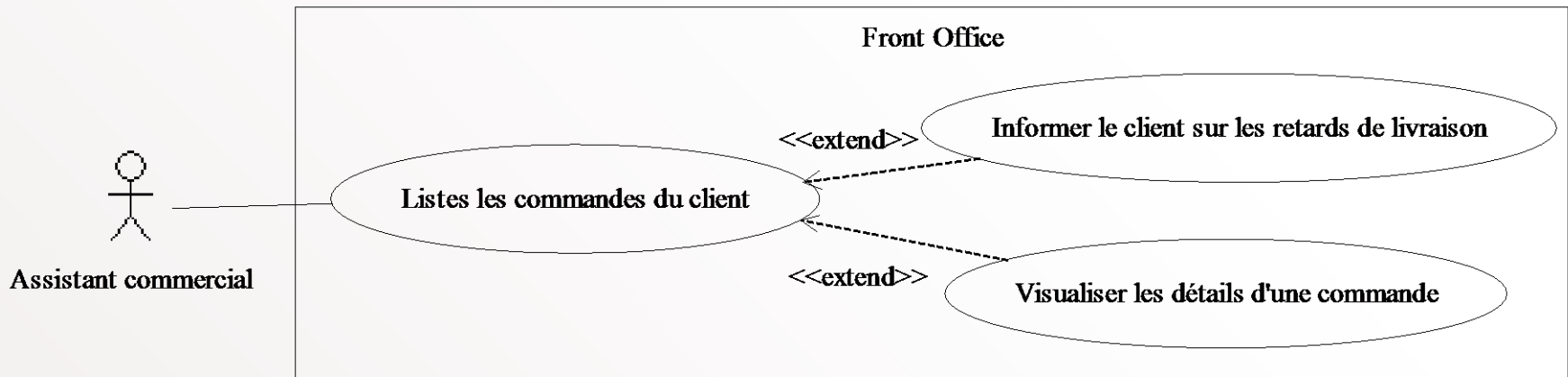


Identification des Packages

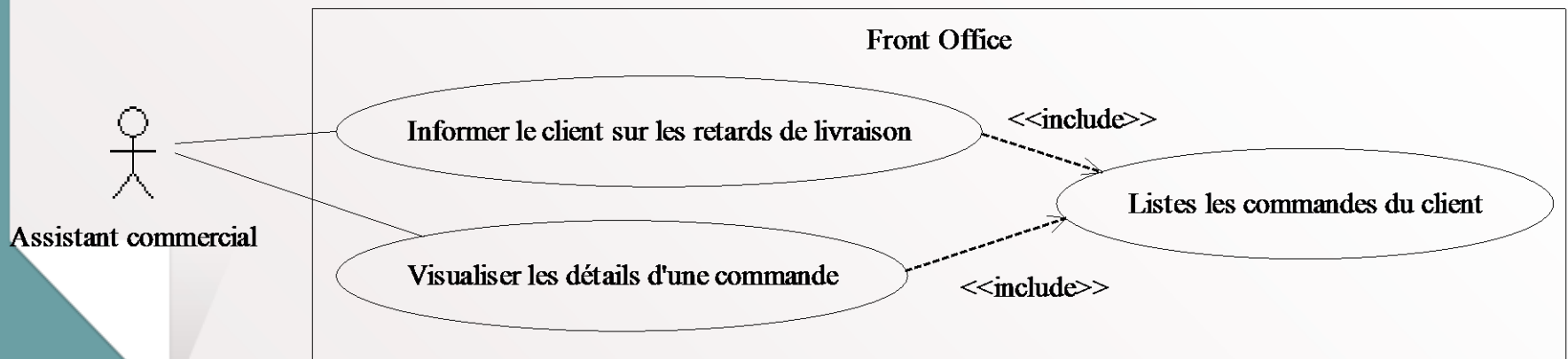
- ❑ Les UC sont structurés en packages selon plusieurs critères possibles:
 - Par acteur
 - Par cohérence fonctionnelle

Erreur classique n°1: représenter la cinématique

Modélisation incorrecte: représente la cinématique de l'application

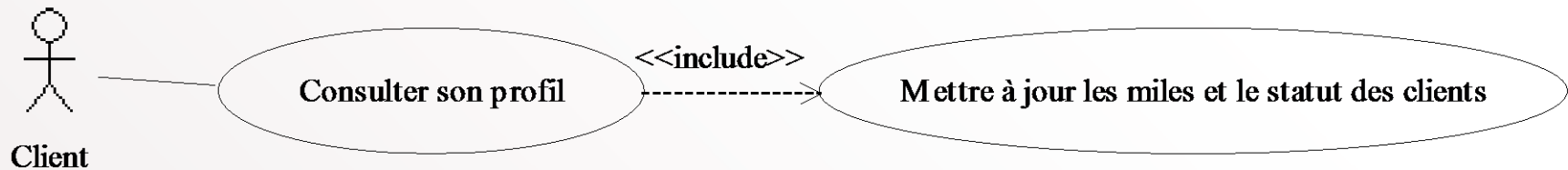


Modélisation correcte: représente les intentions de l'acteur

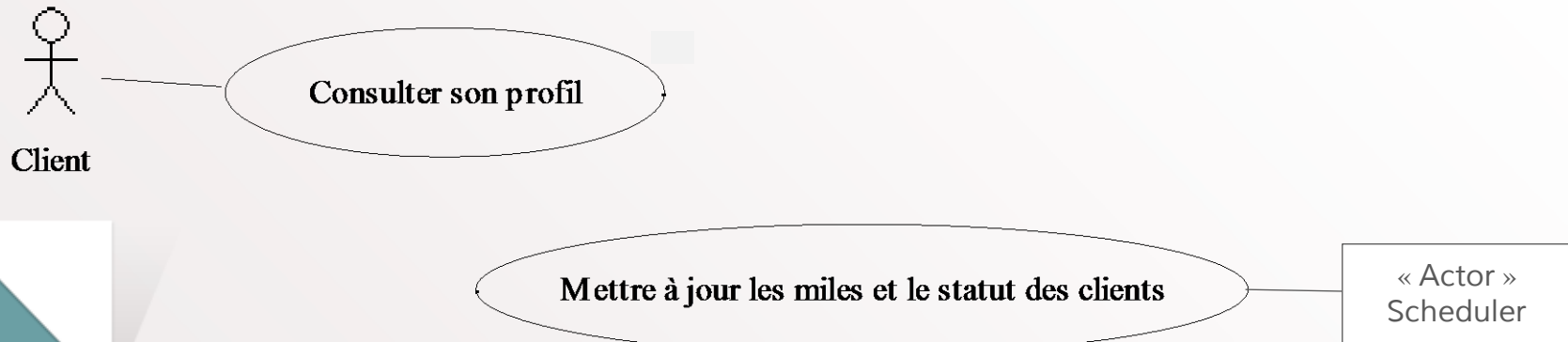


Erreur classique n°2: Confondre l'utilisation des données nécessaires à un UC et leur MàJ

Modélisation incorrecte: les données mises à jour par le UC faussement included sont nécessaires, mais pas l'exécution systématique du UC



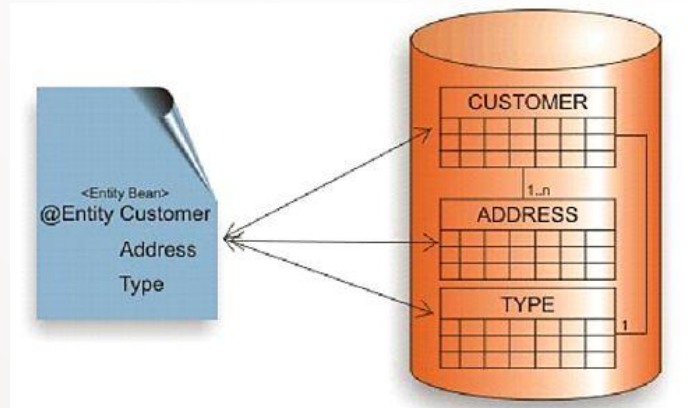
Modélisation correcte: la mise à jour des données et leur consultation sont désynchronisées





Exemple complet: étude de cas Share Trip

3 – Glossaire



Modèle du Domaine

- ❑ Identifier et décrire les concepts d'informations gérés par le S.u.d.
 - Vocabulaire commun
 - Partage des connaissances

- ❑ Favorise la compréhension du métier
 - Nécessaire avant de comprendre le système logiciel

Entité Métier

- ❑ Concept global d'information
 - traduit un choix de gestion pertinent pour le domaine considéré
- ❑ Éléments manipulés dans le cadre d'une activité professionnelle:
 - Eg: commande, facture, contrat, etc...
- ❑ Les entités métier ont des relations entre elles
 - Eg: Une **commande** donne lieu à une **facture**

Identification

❑ Différentes sources:

- Interviews des experts du domaine: utilisateurs ou direction
- Modèle statique établi lors de la réalisation du système existant
- IHM du système existant
- Documents opérationnels issus du système existant

❑ Travail des Business Analysts

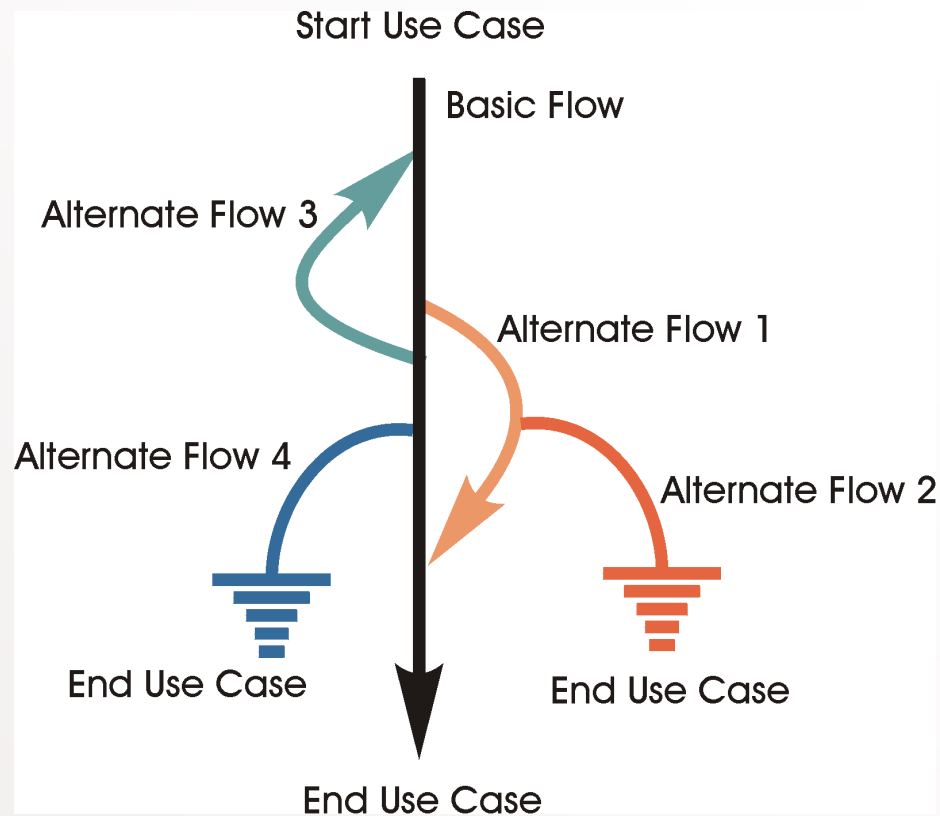
Glossaire

- ❑ Entités Métier: Décrites dans un Glossaire
- ❑ Description textuelle libre. Les informations suivantes peuvent être incluses:
 - Information élémentaires associées
 - Définition précise des informations élémentaires calculées
 - Etats traversés (entités de gestion seulement)
 - Règles de passage d'un état à l'autre (entités de gestion seulement)
 - Liens avec les autres entités

Glossaire et Modélisation

- ❑ Glossaire = travail préparatoire réutilisé en analyse et conception
 - Diagramme de Classes Métier
 - Diagramme d'Etats/Transitions

4 – UC Details



Bordereau d'identification

- ❑ Contient des informations générales à tous les scénarios du UC

Résumé	Le « brief » du UC (déjà décrit dans le UC model).
Déclenchement	Les évènements qui vont déclencher l'exécution du UC.
Objectif	Objectif visé par l'acteur sollicitant le UC.
Fréquence d'utilisation	
Acteurs	Distinguer les acteurs primaires et secondaires s'il y a lieu.
Pre conditions	Etat dans lequel le système doit se trouver avant l'exécution du UC
Post conditions	Idem <i>après</i> l'exécution

Scénarios

- ❑ Chaque UC est ensuite décliné en un ou plusieurs Scénarios:
 - le scénario nominal représente le flot d'évènements qui s'exécute le plus fréquemment.
 - les scénarios alternatifs correspondent à d'autres cas où le UC s'exécute correctement.
 - les exceptions sont des cas où le UC ne s'exécute pas correctement jusqu'au bout.

Flot d'événements

- ❑ Chaque scénario fait l'objet d'une description détaillée:
 - un scénario est une séquence d'actions appelée flot d'évènements
 - ce flot est exprimé textuellement. Il indique ce que fait le système et la façon dont il dialogue avec les acteurs lors de l'exécution du scénario

Lien avec le Glossaire

- ❑ Chaque scénario met forcément en jeu une ou plusieurs entités métier
- ❑ Faire apparaître ces liens entre Glossaire et Flot d'événements
 - Liens hypertexte dans le document

Cf. Exemple de Spécification Fonctionnelle.docx

Précision / Détails

- ❑ Le niveau de détail et la précision requis:
 - dépend du contexte du projet
 - ✓ Développement offshore VS Equipe colocalisée
 - ✓ Complexité du besoin
 - ✓ Besoin de traçabilité (client « polémiste »)
 - ✓ Besoin d'évolutions à court terme
 - peut donc être extrêmement variable d'un projet à l'autre !



❑ Exemple de manque de précision

Version très ambiguë (laisse la place à de multiples interprétations)

Activity
...
L'agent commercial se rend sur le fiche d'activité détaillée du client.
Le système présente la fiche d'activité. Celle-ci contient: <ul style="list-style-type: none"> • Le nom du client • Les ventes du client • ...

Version précise (peut être traduite en requête sql)

Activity
...
L'agent commercial se rend sur le fiche d'activité détaillée du client.
Le système présente la fiche d'activité. Celle-ci contient: <ul style="list-style-type: none"> • Le libellé court de la raison sociale du client • La somme des ventes réalisées auprès du client, sur l'ensemble des produits du catalogue ayant le statut « actif », sur les 3 derniers mois échus • ...

5 – Supporting Requirements

Portability

Ability to easily move the application to a different hardware platform, operating system or even database management system or network protocol

Personalisation

Ability to allow individual users to customise their view of the application/solution (My Yahoo style)

Monitorability

Ability to access information on the applications behaviour

Performance

Throughput, system load, capacity, user volume, response times, transit delay, latency. Possibilities for scheduled processing vs real-time.

Maintainability

Amount of effort required to maintain (and enhance) application/solution in production

Authorisation

Security requirements to ensure users can access only certain functions within the application (by use case, subsystem, web page, business rule, field level etc)

Localisation

Support for multiple languages on entry/query screens in data fields; on reports; multi-byte character requirements and units of measure or currencies

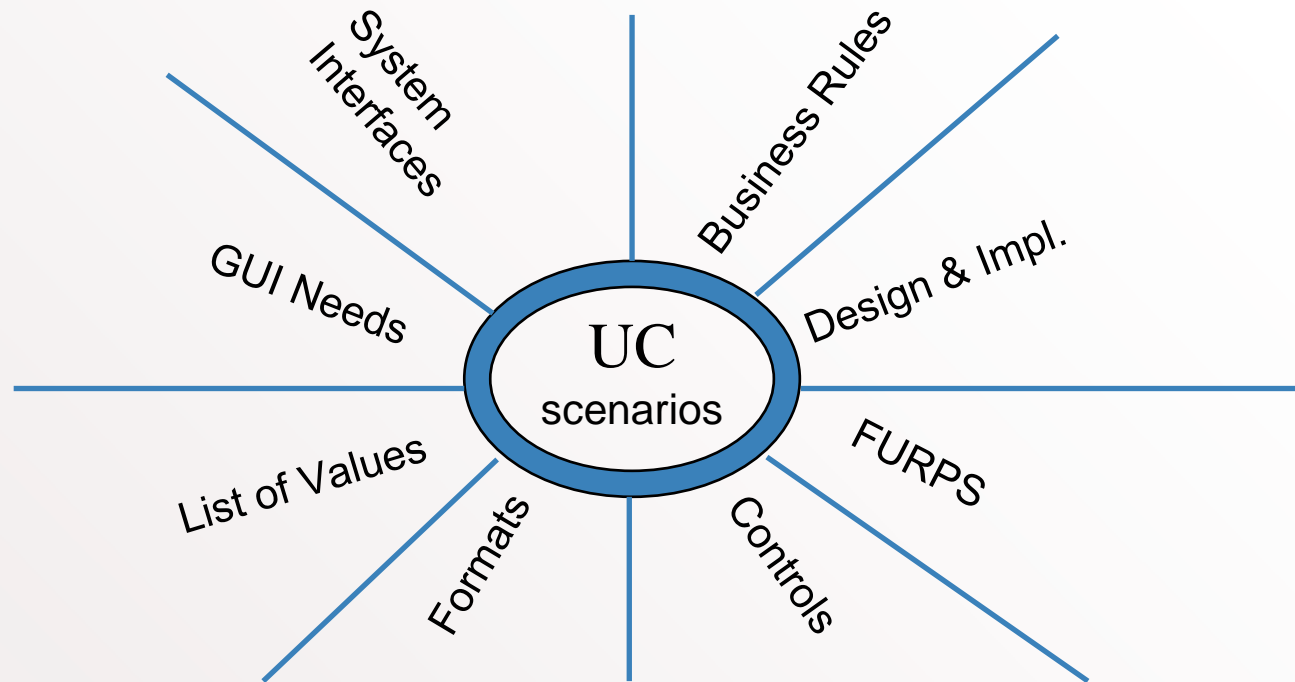
Exprimer TOUT le besoin

- ❑ Exprimer complètement le besoin = spécifier à la fois:
 - Les besoins fonctionnels = scénarios = actions de l'utilisateur et la façon dont le système lui permet d'atteindre l'objectif du UC
 - Les besoins non fonctionnels (BNF) = propriétés supplémentaires du système, très impactants, mais non directement liées à l'objectif du UC



Model « Hub of wheel » (A.Cockburn)

- ❑ Les scénarios constituent le cœur du besoin
- ❑ De nombreux types de besoins s'ajoutent aux scénarios



Exemples simples de BNF

- ❑ Contrainte de performance
 - Temps de réponses

- ❑ Besoin de fiabilité
 - Stress / Accès simultanés
 - Haute disponibilité

- ❑ Apparence globale des IHM

- ❑ Formats – Contrôles + Messages d'erreur



Impact des BNF

- ❑ Les Besoins non Fonctionnels constituent des contraintes du S.u.d.
- ❑ Répondre à ces contraintes nécessite la réalisation de work items qui peuvent être:
 - Très coûteux (AT à dimensionner en conséquence, optimisation du code, nécessité de développer d'autres composants, etc...)
 - Très impactants sur la qualité du produit, voire critiques donc incontournables



□ Exemples

- Performance (temps traitement)

Une application chargée de détecter le franchissement d'un seuil de température sur un réacteur nucléaire. L'alarme doit impérativement se déclencher en moins de 10 ms.

- Fiabilité/Performance (accès simultanés)

Un site de VPC doit être capable de supporter une charge démultipliée autour de Noël et rester toujours accessible sans dégrader ses performances.



Zoom sur la Volumétrie

- ❑ Etablie sur la base d'hypothèses réalistes sur l'activité de l'entreprise (actuelle et/ou cible)
 - Parts de marchés
 - Nombre de clients
 - Nombre de produits vendus
 - Nombre de sites de production
 - ...

- ❑ On en déduit
 - Le nombre de ventes/commandes par an
 - Le nombre de transports de marchandises
 - Le nombre d'accès simultanés au site
 - ...



❏ Exemple simplifié

- Un distributeur de dispositifs médicaux possède 20.000 produits dans son catalogue
- Il vise un marché de 8.000 clients (établissements de santé)
- On suppose que chaque client va passer en moyenne une commande par mois, portant sur 1% du catalogue

→ Chaque mois on enregistre $8 \times (20/100) = 1,6$ Millions de transactions de vente.

→ L'historique sur 5 ans contient 96 M de records

NB: pour plus de réalisme il faudrait tenir compte de la typologie des clients (gros/petits) et de celle des produits (ABC)



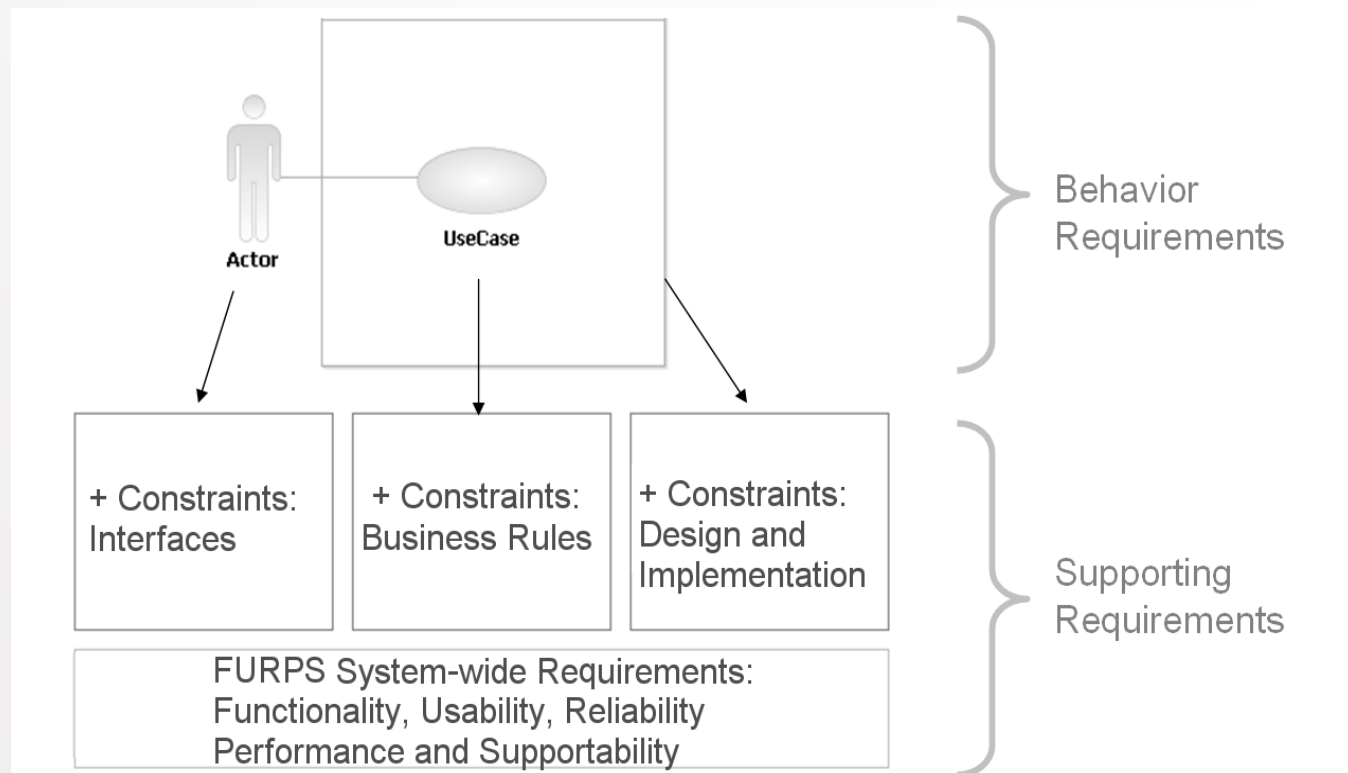
Volumétrie → Contraintes

- ❑ La volumétrie impacte:
 - La performance du S.u.d. : temps de réponse
 - La fiabilité du S.u.d. : nb accès simultanés possibles/scalabilité
 - Les choix liées aux bases de données, à leur mode de stockage, à leur type de traitement

- ❑ Pour tester ces contraintes et faire des démos probantes il faut simuler l'activité
 - ➔ générer des **Mocks** sur les flux et les tables
 - ➔ selon des volumes et distribution de données réalistes



- ❑ Les besoins non fonctionnels sont parfois appelés Supporting Requirements ou « Besoins opérationnels ».
- ❑ Ils peuvent également être classés selon le modèle « FURPS+ »





FURPS (1/2)

☐ Functionality

Fonctionnalités basiques non représentées dans le “UC detail”. Exemples: login, sécurité, impressions, log, etc...

☐ Usability

Facilité d'utilisation (ex : “...approprié pour des utilisateurs non experts de...”), besoin d'aide en ligne, de didacticiel, contraintes d'ergonomie.

☐ Reliability

Capacité du système à fonctionner en conditions de stress, disponibilité (ex : le système sera disponible 24/7), heures limites des traitements batch, tolérance de pannes, recoverability, modes de fonctionnement dégradé admissibles..

FURPS (2/2)

☐ Performance :

Temps de réponse, volumétrie (ex : « ...2000 accès simultanés.. »), fréquence d'utilisation, ...

☐ Supportability :

Facilité de maintenance du système (ex : "...toutes les données seront accessibles via des tables administrateur et modifiables sans recompilation..."), évolutivité, scalabilité, compatibilité, exigences d'installation, niveau de support, ...