

## IFT 2255 - Genie Logiciel

# Exigences

# Terminologie

- **Besoin:** ce qui est nécessaire pour que le client puisse résoudre son problème
- **Exigence (*requirement*):** expression d'un besoin documenté sur ce que le système doit faire
- **Spécifications:** ensemble des exigences explicites à satisfaire, prérequis pour le développement du système
- En pratique, *besoin* = *exigence* = *spécification*

# QUESTION

*D'où proviennent les besoins?*

- Besoins initiaux, ex: système de contrôleur d'ascenseur
  1. Le programme doit planifier les activités de l'ascenseur de façon efficace et raisonnable
  2. Le programme doit illuminer l'indicateur du panneau d'arrivée correspondant à l'étage où l'ascenseur arrive
  3. Au dernier (resp. premier) étage, le panneau d'appel ne contient qu'un seul bouton, soit celui pour descendre (resp. monter)
- Besoins ajoutés durant le processus de développement
  1. Bogue rapporté par un utilisateur
  2. Utilisateur demande une nouvelle fonctionnalité (fermer la porte sur demande, la maintenir ouverte...)
  3. Programmeur propose une amélioration (et si...?)
  4. Gestionnaire désire rendre un produit concurrentiel

# Identification des besoins

- Identifier les besoins correctement est difficile!
- Causes les plus répandues de l'échec de projets
  - Contribution insuffisante du client
  - Spécification incomplète des besoins
  - Changement des besoins
- Difficultés :
  - Problèmes complexes
  - Connaissance limitée du domaine
  - Clients sans connaissances techniques
  - Communication difficile

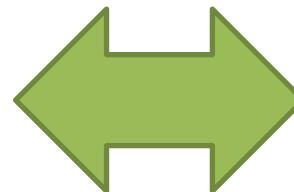
# Le rôle de l'analyste



Client



Analyste



Développeur

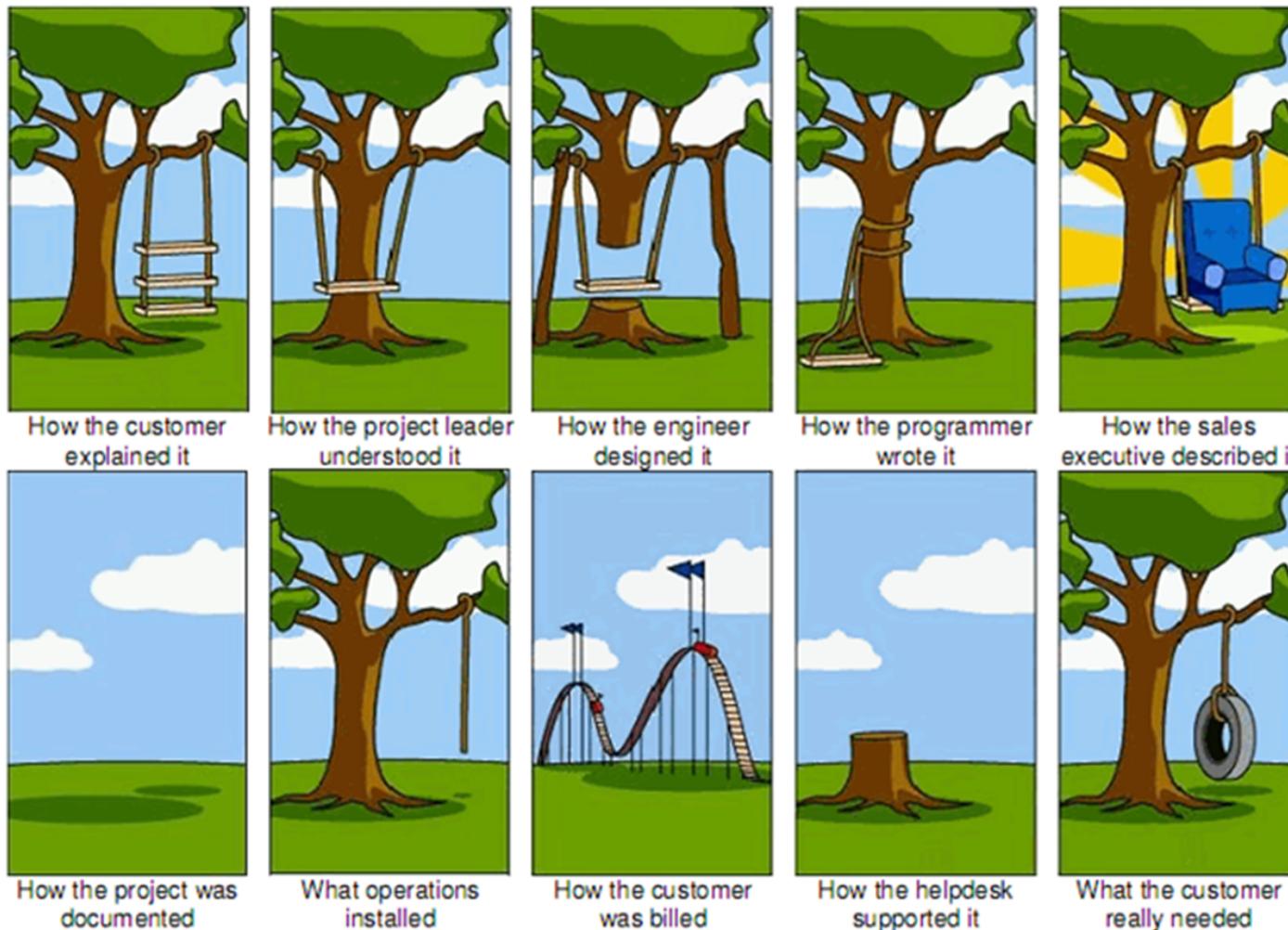
- Expertise, jargon du domaine
- Indécis, son opinion change selon l'offre
- Besoins ambigus, éléments manquants

- Doit devenir aussi informé du fonctionnement de l'entreprise que les utilisateurs
- Doit devenir l'expert

- Schémas, langages formels
- Spécifications souvent incompréhensibles pour les non initiés

# But du workflow des exigences

Déterminer ce dont le client A BESOIN, plutôt que ce qu'il VEUT



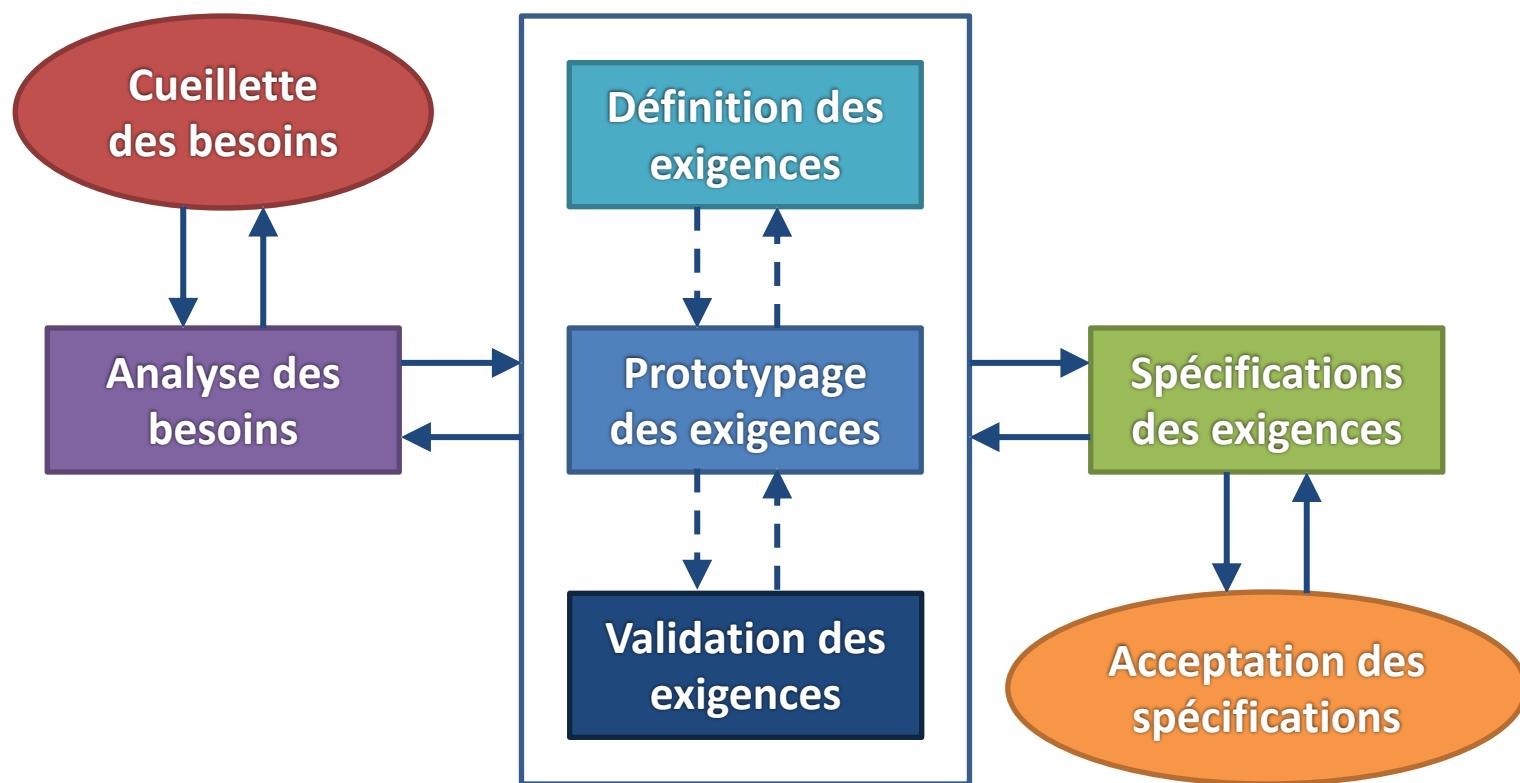
# Solution

1. Comprendre le **domaine** d'application
  - Environnement spécifique au métier où le logiciel sera exploité
2. Construire un **modèle d'affaire**
  - Processus d'affaire, ce qui est informatisable, mode d'opération
3. Utiliser le modèle d'affaire pour déterminer les **exigences** du client

Faire plusieurs **itérations** jusqu'à ce que les exigences soient satisfaisantes

# Ingénierie des exigences

*Déf :* Ensemble des activités reliées au développement et à l'entente des spécifications finales.



# Cueillette des besoins

- **Rencontre** initiale avec le client
  - Représentant des ventes, analyste d'affaires, architecte logiciel
- Client expose le produit **conceptuel** qu'il veut
  - Description vague, pas raisonnable, contradictoire, impossible à faire
- C'est à vous de déterminer exactement ce qu'il veut et découvrir quelles sont les **contraintes**

# Cueillette des besoins

## Comprendre le domaine

- Identification des opérations et procédés administratifs
  - « Que faites-vous ? »
- Réalisation des opérations
  - « Comment le faites-vous ? »
- Identification des informations requises pour réaliser les opérations
  - « Quelles informations utilisez-vous ? »
- Construire un **glossaire** avec la bonne terminologie
  - Liste de mots techniques utilisés dans le domaine avec leur sens

# Cueillette des besoins

## Établissement des contraintes

- Calendrier du projet
  - Habituellement en mois
  - Souvent des logiciels critiques et essentiels à l'entreprise
- Coût
  - Client n'informe pas le développeur du vrai budget disponible
  - Procédure d'appel d'offre
- Opération en parallèle
- Fiabilité du logiciel
- Mobilité du logiciel
- Temps de réponse rapide

# QUESTION

*Comment recueillir les besoins du client ?*

- Entrevues
- Sondage
- Observation
- Étude de documents existants

# Méthodes pour collecter l'information

## Entrevues

- Entrevue avec clients, utilisateurs et experts du domaine
  - Entrevue **structurée**: formelle, questions préparées à l'avance, souvent des questions fermées
  - Entrevue **non structurée**: informelle, questions posées d'après les réponses reçues, souvent des questions ouvertes
- Types de questions
  - **Fermée**: requiert une réponse spécifique
  - **Ouverte**: encourage la personne à parler et donner des explications supplémentaires
  - **Choix multiple**: restreint les réponses possibles
  - **Échelle de Likert**, ex: d'accord, pas d'accord, etc.



# Méthodes pour collecter l'information

## Questionnaire

**Questionnaire de RMO**

Ce questionnaire est envoyé à tout le personnel du service des ventes par téléphone. Comme vous le savez, RMO est en train de développer un nouveau système de soutien des clients pour la prise des commandes et le service à la clientèle.

Ce questionnaire a pour but d'obtenir de l'information préliminaire pour nous aider à définir les spécifications du nouveau système. Des discussions de suivi seront organisées par la suite pour permettre à tous d'élaborer sur ces spécifications.

**1<sup>re</sup> partie. Répondez à ces questions fondées sur un quart de travail de quatre heures.**

1. Combien d'appels téléphoniques recevez-vous?
2. Combien d'appels téléphoniques faut-il pour passer une commande?
3. Combien d'appels ne sont que des questions sur les produits de RMO?
4. Évaluez combien de fois pendant votre quart de travail les clients vous demandent des articles en rupture de stock?
5. De ces demandes d'articles épuisés, quel pourcentage sera placé sur une liste des commandes en souffrance?
6. Combien de fois un client passe-t-il une commande à partir d'un catalogue périmé?
7. Combien de fois un client annule-t-il une commande au milieu de la conversation?
8. Combien de fois des commandes sont-elles refusées pour cause de mauvais crédit?

**2<sup>re</sup> partie. Sur l'échelle de 1 à 7 qui suit, encerclez le chiffre correspondant à votre accord ou votre désaccord avec l'énoncé.**

Question	Fortement d'accord	Fortement en désaccord					
	1	2	3	4	5	6	7
Je pourrais mieux faire mon travail si je pouvais donner des descriptions plus longues des produits aux clients à qui je parle.	1	2	3	4	5	6	7
Je pourrais mieux faire mon travail si j'avais accès à l'historique d'achat du client.	1	2	3	4	5	6	7
Je pourrais mieux servir les clients si j'avais accès à de l'information sur les accessoires adéquats pour les articles commandés.	1	2	3	4	5	6	7
Le temps de réponse de l'ordinateur est lent et me cause des difficultés lorsque je réponds aux demandes des clients.	1	2	3	4	5	6	7

**3<sup>re</sup> partie. Veuillez inscrire vos opinions et commentaires.**

Identifiez brièvement les problèmes du système actuel que vous aimeriez voir résolus par un nouveau système.

---



---



---

## Questionnaire

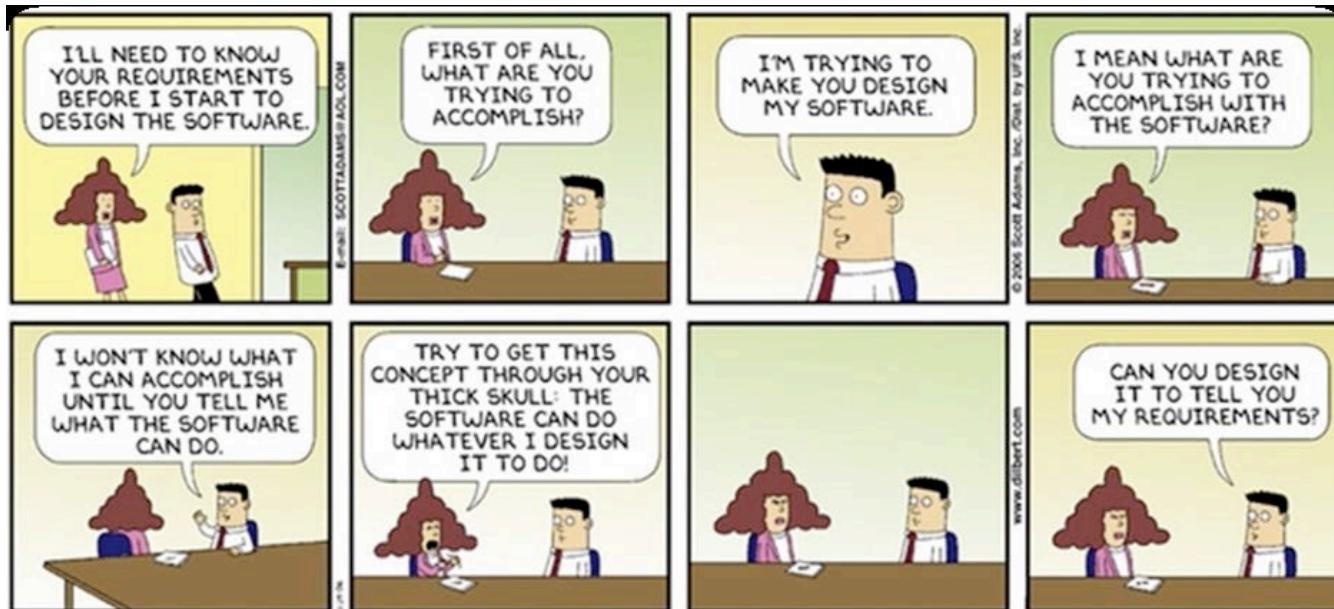
### Questions fermées objectives

### Questions fermées subjectives

### Questions ouvertes subjectives (explicatives)

# Difficulté d'une entrevue

- Ce n'est pas facile de faire une entrevue
  - Trop structurée  $\Rightarrow$  pas assez d'infos
  - Pas assez structurée  $\Rightarrow$  information trop vague
  - L'intervieweur doit être familier avec le domaine
  - Il doit rester ouvert d'esprit en tout temps
- Après l'entrevue, préparer un rapport écrit



# Autres méthodes pour collecter l'information

- **Observation**
  - **Passive**: examiner comment le client et les utilisateurs potentiels travaillent dans l'organisation
  - **Active**: participer aux activités de l'organisation
  - En personne, ou en enregistrant
    - Problème de confidentialité des employés
- **Étude de documents et logiciels existants**
  - **Documents d'entreprise**: procédures de travail, plans, minutes de réunion, description d'emploi, correspondances, etc.
  - **Formulaire et rapports**: rapports techniques, saisie d'écran, manuels d'utilisateur, modèle d'analyse et de conception, etc.
  - Journaux d'affaires, livres de gestion
- **Étude des solutions (déjà existantes) de fournisseurs**

# Prototypage des exigences

- Maquette démonstrative, première étude de faisabilité
- Identification de besoins conflictuels, omis ou mal saisis
- Prototype **jetable**
  - Porter attention sur les besoins moins bien compris
- Prototype **évolutif**
  - Porter attention sur les besoins les mieux compris
- Développement **conjoint** d'applications
  - Ateliers et réunions auxquelles les clients et les développeurs participent
  - Accélère les communications et le feedback
  - Réduit les risques de manifester une nouvelle exigence

# Exigences de haut-niveau

- **Business case:** document exécutif, contraintes sur le logiciel et le projet
- **Opportunité et besoin**
  - Système d'inventaire perd 50% des commandes; il y a 2M\$ d'inventaire en surplus;
- **Justification du projet**
  - **Cadre**
    - Contrôle de l'inventaire et du traitement des commandes
  - **Contraintes majeures**
    - Budget, temps, risques
  - **Fonctionnalités majeures**
    - Contrôle d'inventaire amélioré en automatisant le traitement des commandes et livraisons; client commande en ligne; gestion de livraison en ligne
  - **Facteur de réussite**
    - Doit réduire l'inventaire; ne doit perdre aucune commande
- **Caractéristiques des utilisateurs**
  - Qui va être affecté au produit final?
  - Qualifiés, besoin d'entraînement

# Exigences détaillées

Besoins fonctionnels

Flux d'affaires

Données et formats

Interfaces utilisateur

Interface avec d'autres systèmes

Contraintes non fonctionnelles

# Besoins fonctionnels

- Groupe évident, point de départ
  - Que doit **faire** le système ?
    - « Le système lit les fiches des employés et imprime des chèques de paie. »
- Comportement, caractéristiques et capacités du système
- Spécifie une action que le système doit accomplir sous forme d'interaction entre le **système** et son **environnement**
  - Exprimé en terme d'entrées et de sorties
  - Indépendant de l'implémentation
- Généralement traités durant les workflows d'exigences et d'analyse

# Flux d'affaires

- Fonctionnalité doit être décrite dans le contexte d'un flux d'affaire
  - Comment est-ce que ça affecte les tâches de l'utilisateur ?
- Scénarios étape par étape
- Cas d'utilisation
  - Séquence d'actions que le système doit effectué dans le contexte du flux d'affaire de l'utilisateur

# Données et formats

- Détermine les données en entrée et sortie de l'application
  - Qu'est-ce qui doit être entré dans le système ? Dans quel but ?
    - « L'utilisateur entre la date sous forme DD-MM-YYYY. »
- Certaines données en entrée peuvent déclencher un processus
- La sortie peut être sous forme d'une réponse à une requête ou un rapport
- Format permis de l'information entrée
- Format requis de l'information sortie
- Messages d'erreur, alertes, texte d'aide
- Plurilinguisme

# Interface utilisateur

- Comment les entrées et sorties sont représentées
  - Présentation, thème
    - Graphique
  - Mode
    - En ligne
- Flux d'utilisation doit suivre le flux d'affaire
- Captures d'écran, maquettes, prototypage rapide



# Interface avec d'autres systèmes

- Applications existantes, systèmes dans le réseau
- Certains besoins peuvent nécessiter de modifier comment les utilisateurs opèrent des systèmes adjacents
- Dimensions à considérer
  - Transfert de contrôle
  - Transfert de données
  - Réception de réponses
  - Reprise d'erreur, capacité de réessayer, messages

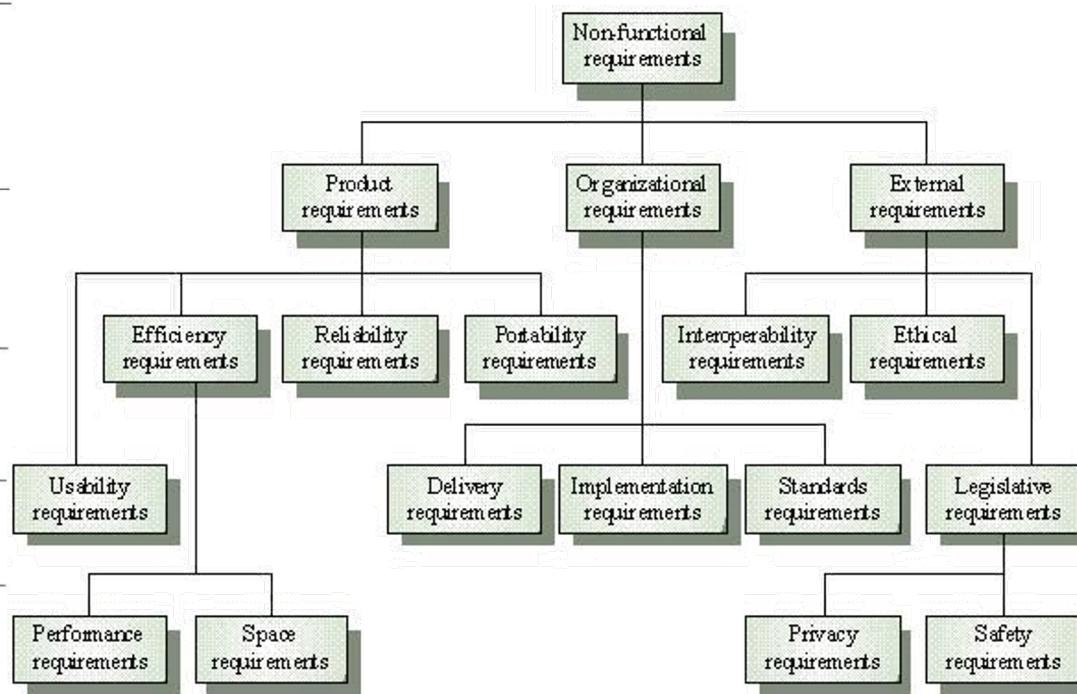
# Besoins non fonctionnels

- Spécifie des propriétés du logiciel même
  - Contraintes sur la plateforme, temps de réponse, performance, fiabilité, sécurité, adaptabilité, disponibilité, maintenabilité, etc.
- Généralement attendre le workflow de conception
  - Impacte l'architecture du système
- Contraintes
  - Langages de programmation, système d'exploitation, outils autorisés, licences

# Questions pour besoins non fonctionnels

Table 4-3 Example questions for eliciting nonfunctional requirements.

Category	Example questions
Usability	<ul style="list-style-type: none"> <li>• What is the level of expertise of the user?</li> <li>• What user interface standards are familiar to the user?</li> <li>• What documentation should be provided to the user?</li> </ul>
Reliability (including robustness, safety, and security)	<ul style="list-style-type: none"> <li>• How reliable, available, and robust should the system be?</li> <li>• Is restarting the system acceptable in the event of a failure?</li> <li>• How much data can the system loose?</li> <li>• How should the system handle exceptions?</li> <li>• Are there safety requirements of the system?</li> <li>• Are there security requirements of the system?</li> </ul>
Performance	<ul style="list-style-type: none"> <li>• How responsive should the system be?</li> <li>• Are any user tasks time critical?</li> <li>• How many concurrent users should it support?</li> <li>• How large is a typical data store for comparable systems?</li> <li>• What is the worse latency that is acceptable to users?</li> </ul>
Supportability (including maintainability and portability)	<ul style="list-style-type: none"> <li>• What are the foreseen extensions to the system?</li> <li>• Who maintains the system?</li> <li>• Are there plans to port the system to different software or hardware environments?</li> </ul>
Implementation	<ul style="list-style-type: none"> <li>• Are there constraints on the hardware platform?</li> <li>• Are constraints imposed by the maintenance team?</li> <li>• Are constraints imposed by the testing team?</li> </ul>
Interface	<ul style="list-style-type: none"> <li>• Should the system interact with any existing systems?</li> <li>• How are data exported/imported into the system?</li> <li>• What standards in use by the client should be supported by the system?</li> </ul>
Operation	<ul style="list-style-type: none"> <li>• Who manages the running system?</li> </ul>
Packaging	<ul style="list-style-type: none"> <li>• Who installs the system?</li> <li>• How many installations are foreseen?</li> <li>• Are there time constraints on the installation?</li> </ul>
Legal	<ul style="list-style-type: none"> <li>• How should the system be licensed?</li> <li>• Are any liability issues associated with system failures?</li> <li>• Are any royalties or licensing fees incurred by using specific algorithms or components?</li> </ul>



# Validation et négociation

- Les besoins répondent-ils aux exigences du client ?
- Réviser la liste des besoins
  - Sont-ils cohérents? Clairs? Intelligibles?
  - Élimination des besoins non pertinents ou irréalistes
    - Bien définir les frontières du système
    - Identifier les besoins qui ne répondent pas aux objectifs du système, qui sont hors-plan
  - Faire la liste des besoins exclus pour cause de
    - trop grande difficulté de réalisation
    - mise en œuvre par matériel (*hardware*)
    - technologie existante inadéquate
- Tout compromis doit être négocié avec le client

# Validation et négociation

- Classer les besoins selon leur priorité et évaluer le risque associé à chacun
- Dans un contexte agile, re-prioriser les exigences dans le carnet du produit pour le prochain sprint
- Quels sont les besoins susceptibles de causer des problèmes pendant le développement ?
  - Risques techniques
  - Risques de performance
  - Risques politiques, légaux, éthiques
  - Risques de volatilité

# Priorisation des besoins

## Par valeur d'affaire

1. Fonctionnalité essentielle
2. Fonctionnalité nécessaire
3. Fonctionnalité importante
4. Amélioration majeure
5. Amélioration mineure

# Priorisation des besoins

## Par sévérité du risque

1. Risque sévère
2. Risque important
3. Risque éloigné
4. Obstacle mineur

# Priorisation des erreurs

## Par sévérité

1. Erreur fatale
2. Application sévèrement compromise
3. Certaines fonctionnalités compromises
4. Problème mineur

# Livrables du workflow des exigences

- Business case
- Cahier des charges
- Cas d'utilisation

Le tout approuvé par le client

# Cahier des charges

- 1. Description générale du projet
  - 1. Intention et portée du projet
  - 2. Contexte d'entreprise (planification stratégique)
  - 3. Parties prenantes
  - 4. Idées de solution
  - 5. Plan du document
- 2. Services du système
  - 1. Portée du système (diagramme de contexte)
  - 2. Besoins fonctionnels
  - 3. Besoins des données (attributs, interrelations)
- 3. Contraintes du système
  - 1. Contraintes d'interface
  - 2. Contraintes de performance
  - 3. Contraintes de sécurité
  - 4. Contraintes opérationnelles
  - 5. Contraintes politiques et légales
- 4. Éléments du projet
  - 1. Problèmes ouverts
  - 2. Planning préliminaire
  - 3. Budget préliminaire
- 5. Appendices
  - 1. Glossaire
  - 2. Documents et formulaires d'entreprise
  - 3. Références bibliographiques