

# Projet dirigé

Étude des besoins

# Étude des besoins

## Plan de la séance

- Retour sur la séance précédente
  - Point de vue de l'enseignant
  - Point de vue de l'étudiant
- Objectifs de l'étude des besoins
- Recueillir les besoins
- Formaliser les besoins
- Les cas d'utilisation: formalisme et description
- Description des cas d'utilisation

# Étude des besoins: Objectifs

COMPRENDRE  
CE QUE LE  
CLIENT VEUT

ÉTABLIR LA  
FAISABILITÉ DU  
PROJET

FAIRE DES  
PROPOSITIONS  
DE SOLUTIONS

# Étude des besoins, problèmes

Pourcentage de fonctionnalités implémentées réellement utilisées

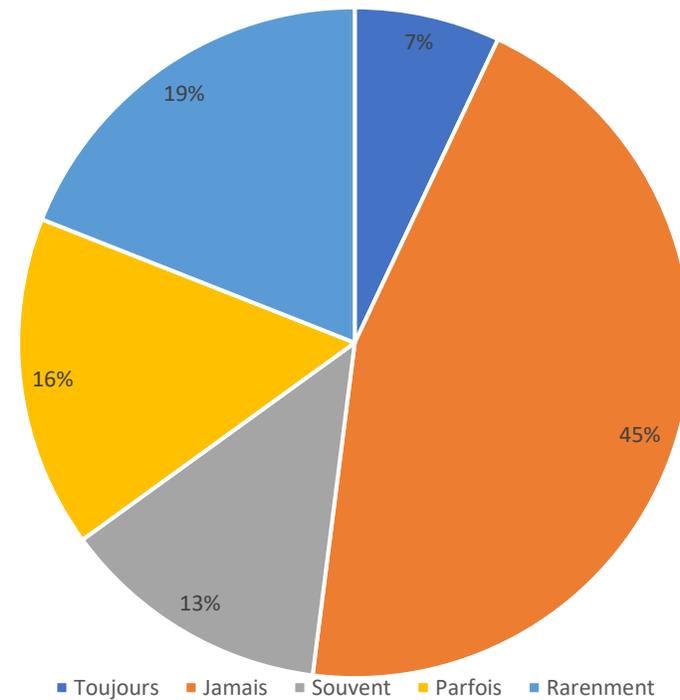


Figure 1, (Source : Gestion de projet agile (3e Edition, Véronique Messager Rota)

# Étude des besoins, problèmes

## Origines des défauts logiciels

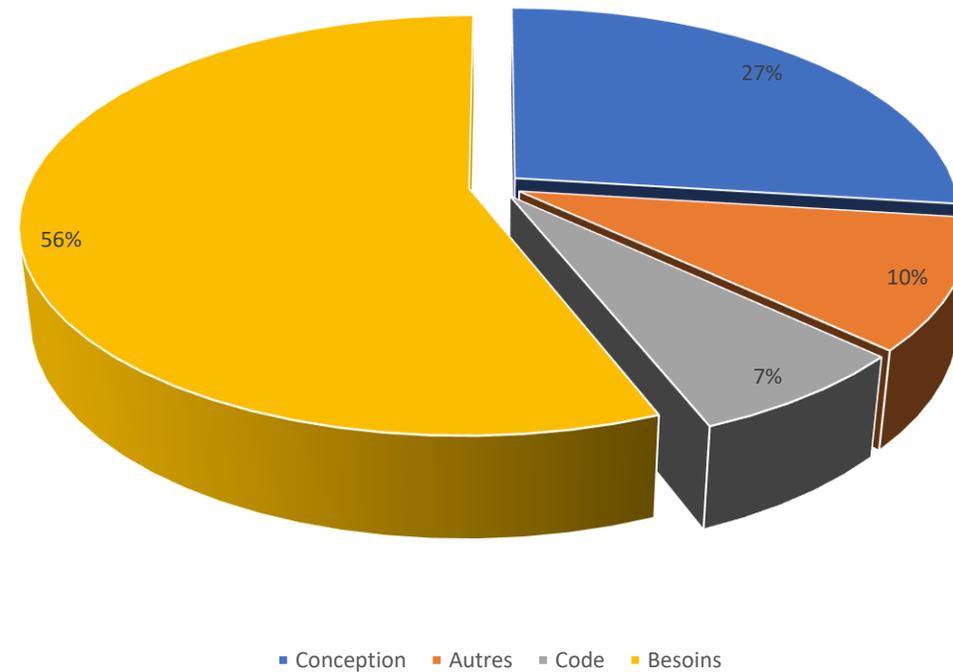


Figure 2, Origines des défauts logiciels (Source : Gestion de projet agile (3e Edition, Véronique Messenger Rota)

# Étude des besoins, problèmes

- Comment est-il possible de développer des fonctionnalités qui ne seront jamais utilisées ?
- Est-il possible que nous ayons mal communiqué avec le client ?
- Est-il possible que nous ayons mal compris les besoins du client?
- Le client a-t-il bien exprimé ses besoins ?
- Les attentes du client sont-elles clairement exprimées

De la difficulté à recueillir les besoins (on se rappelle l'exercice: Le Puzzle)

# Recueillir les besoins

## **Partager une vision**

La vision d'un produit ou d'un projet est l'orientation générale donnée à l'équipe, l'objectif global à atteindre. Sans objectif, sans vision l'équipe de projet ne va nulle part. « *il n'est pas de vent favorable à celui qui ne sait où il va* », Sénèque

## **Faire émerger les besoins de manière itérative et incrémentale car:**

- Entre la représentation qu'a le client de son futur produit, la difficulté parfois à le décrire, l'interprétation possible de cette description par le l'équipe de réalisation et le produit qui est livré au final, il y a de nombreux risques de perdre le besoin initial.
- Une idée initiale peut s'avérer inutile, trop coûteuse, ou trop « secondaire » après analyse.
- Tous les besoins n'ont pas la même priorité.
- Le client pourrait introduire une autre demande, renoncer à une demande.

# Recueillir les besoins

## **Recueillir les besoins : comment ?**

- Brainstorming
- L'interview
- L'observation
- Le questionnaire
- L'analyse de l'existant

Faire impliquer le client le plus souvent possible car les besoins changent, les besoins évoluent. (exemple: ce qui nous semblait important au début, est finalement peu important ou trop coûteux à réaliser)

Ne pas oublier des acteurs dans le système (le puzzle pour lequel il manquerait une pièce)

# Formaliser les besoins

## **Formaliser les besoins : Pourquoi ?**

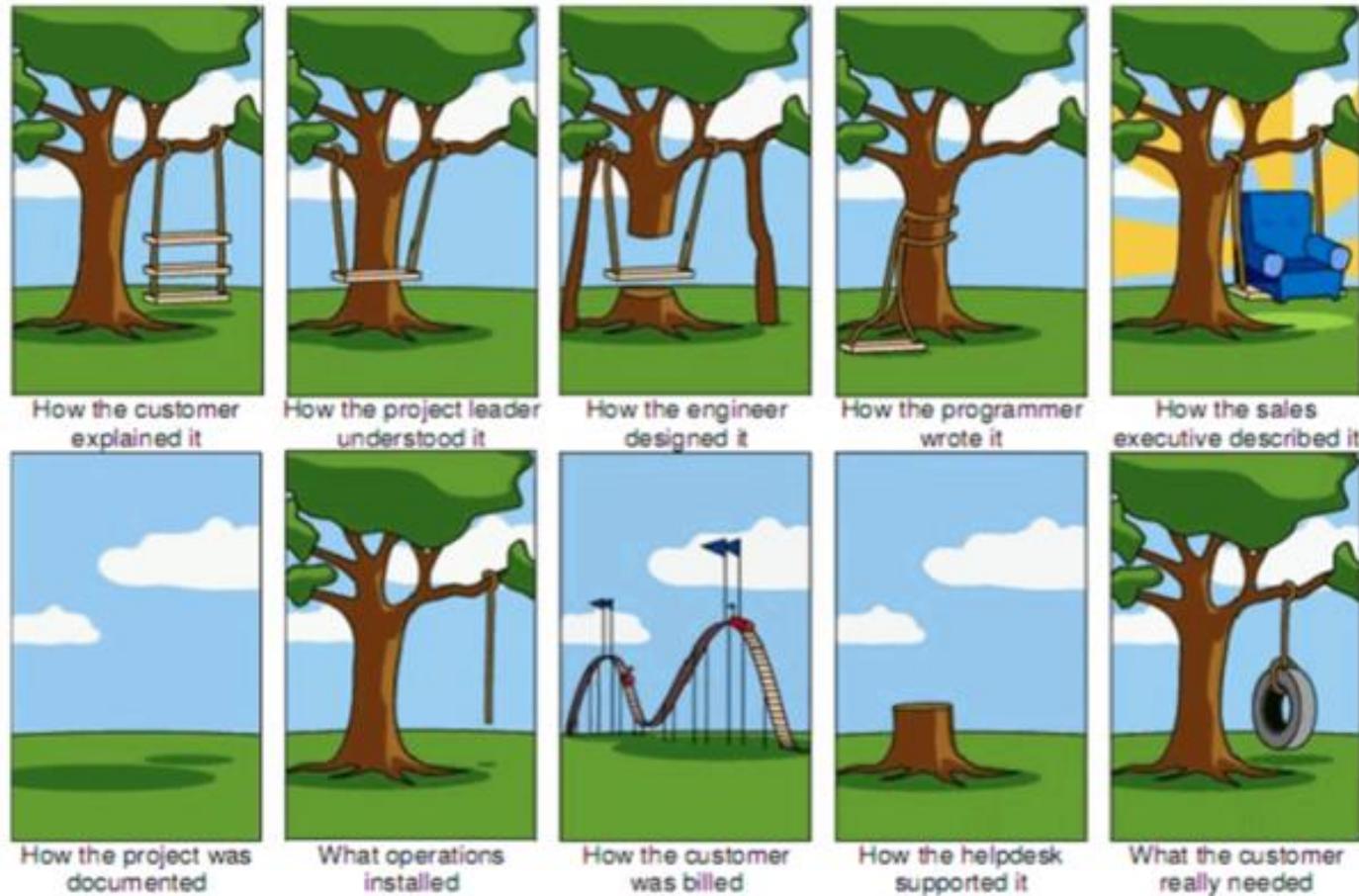
Car on veut retrouver la trace des besoins que nous venons de recueillir. On voudra les consulter, les mettre à jour etc..

## **Formaliser les besoins : Comment ?**

Comment garder un lien entre les besoins initiaux et les besoins des étapes intermédiaires ?  
Comment s'assurer que tous les besoins sont traités ? En d'autres mots, comment limiter la rupture dans le processus de développement ?

Voir image suivante.

# Formaliser les besoins



# Formaliser les besoins: norme IEEE830

*Institute of Electrical and Electronics Engineers*

## Formaliser les besoins : Comment ?

- **En utilisant la norme IEEE 830.** → on parle de SEL (spécification des exigences logiciel)
  - Doit être exprimée de manière claire, concise et cohérente
  - Doit être non ambiguë
  - Doit être valide (une exigence réelle)
  - Doit avoir un bénéfice qui l'emporte sur le coût qu'elle engendre
  - Doit être vérifiable
  - Doit être identifiable de manière unique
  - Doit être modifiable (car elle évolue)
  - Doit être importante dans la résolution du problème
  - Doit être réaliste en considérant les ressources disponibles.
  - Doit être exprimée d'une manière concise et cohérente

Exemple: la fonctionnalité F01, sera énoncée ainsi:

F01: À la demande de l'étudiant, le système affichera son bulletin.

# Formaliser les besoins: les cas d'utilisation

- **En utilisant :Les use-case ou les cas d'utilisation d'UML (Unified Modeling Langage)**

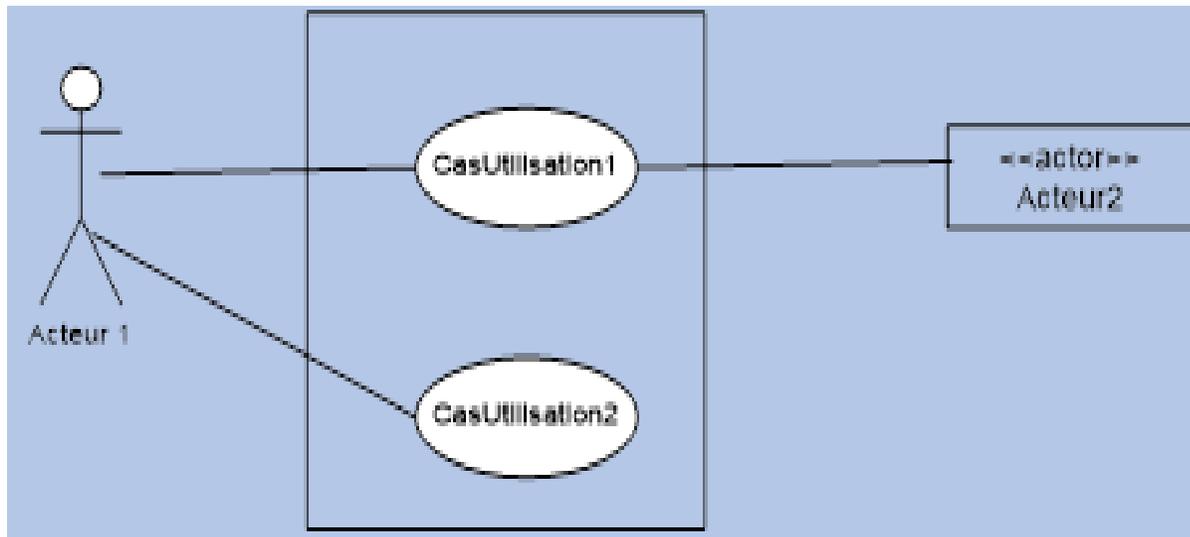
Les cas d'utilisation est une technique de formalisation des besoins utilisée dans le processus unifié. Le diagramme de cas d'utilisation d'UML est utilisé pour représenter les besoins du système (ce que le système doit faire). Ils décrivent le comportement du système du point de vue de l'utilisateur.

- Permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- Ils centrent l'expression des exigences du système sur ses utilisateurs ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.
- Ils identifient les utilisateurs(acteurs) du système et leur interaction avec celui-ci

# Formaliser les besoins: les cas d'utilisation

**En utilisant :Les use-case ou les cas d'utilisation d'UML (Unified Modeling Langage)**

Le diagramme des use-case permet, **d'un seul coup d'œil**, de voir les utilisateurs du système et leur interaction avec lui (système)



# Formaliser les besoins: les cas d'utilisation

- **En utilisant :Les use-case ou les cas d'utilisation d'UML (Unified Modeling Langage)**

Comment faire un diagramme des use-case ?

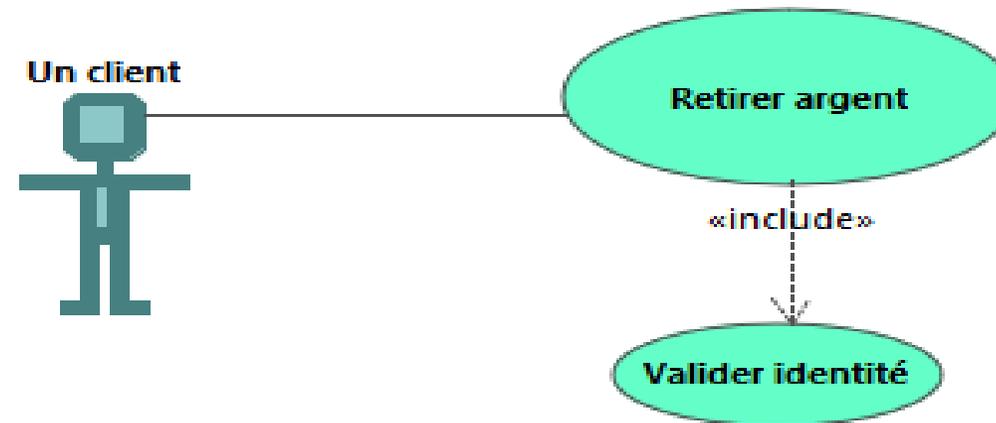
1. Recenser **tous les acteurs** du système
2. Comprendre ce que les acteurs font dans le système ou avec le système
3. Un cas d'utilisation n'est pas une action à faire. Un cas d'utilisation **est un ensemble de séquences d'actions** qui sont réalisées par le système et qui produisent un résultat observable pour un acteur particulier du système
4. Déterminer les cas principaux en premier. Toujours revenir au BUT initial.
5. Déterminer les liens entre les cas d'utilisation.
6. Si un diagramme de cas d'utilisation est trop gros...(trop de cas) il perd de son objectif.

# Formaliser les besoins: les cas d'utilisation

Relation entre cas d'utilisation: La relation « include » ou « uses » (utilise)

Un cas A **utilise** un cas B si les actions de A ne peuvent s'exécuter avant l'exécution des actions de B. En d'autres mots, pour exécuter A il faut d'abord exécuter B.

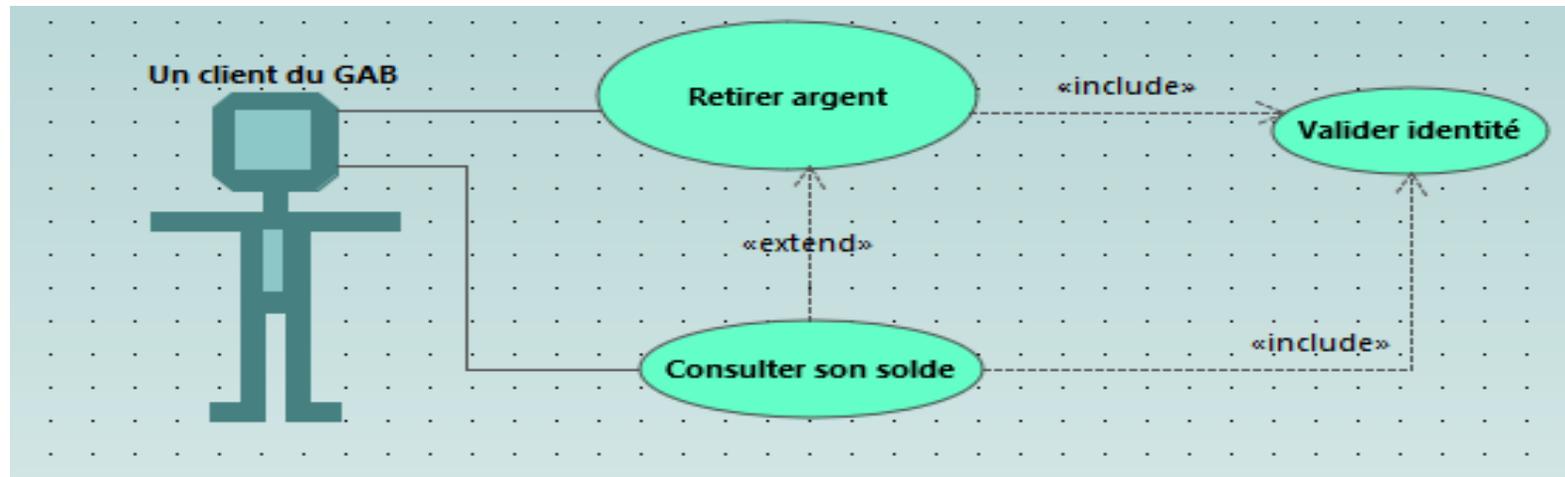
Exemple, un client ne peut retirer de l'argent sans s'identifier



# Formaliser les besoins: les cas d'utilisation

Relation entre cas d'utilisation: La relation « extend » ou Étend.

Un cas A **étend** son action sur un autre cas B si B est la suite logique de A. De plus A et B pourrait s'exécuter de manière indépendante.



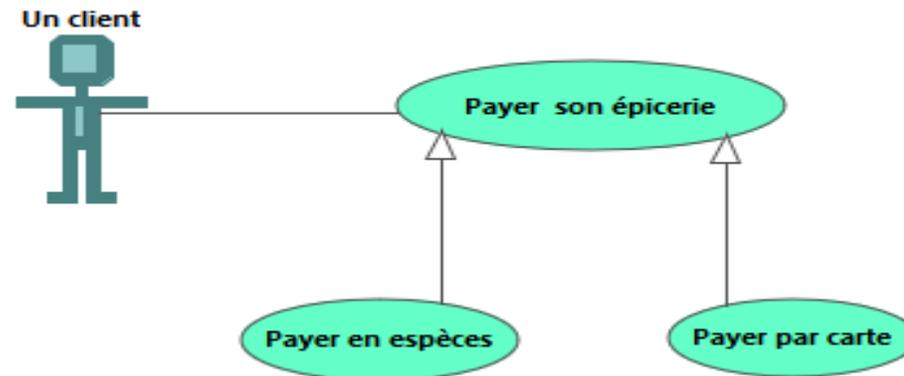
# Formaliser les besoins: les cas d'utilisation

- Les cas d'utilisation ***Retirer argent*** et ***Consulter son solde*** sont deux cas indépendants, en ce sens qu'un client du GAB peut consulter son solde sans retirer de l'argent ou qu'il peut retirer de l'argent sans effectuer d'interrogation de solde.
- Cependant le cas ***Consulter son solde*** peut étendre son action au cas ***Retirer de l'argent***. Un client après avoir consulter son solde, peut décider de retirer de l'argent.
- Quelque soit le cas d'utilisation à exécuter (***Retirer argent*** ou ***Consulter son solde***), ils doivent faire appel (utilise) au d'utilisation Identifier *le client* (carte débit et nip).

# Formaliser les besoins: les cas d'utilisation

Relation entre cas d'utilisation: La relation de généralisation

- Un cas A est une généralisation d'un cas B si B est un cas particulier de A
- Cette relation de généralisation/spécialisation est présente dans la plupart des diagrammes UML et se traduit par le concept d'héritage dans les langages orientés objet.

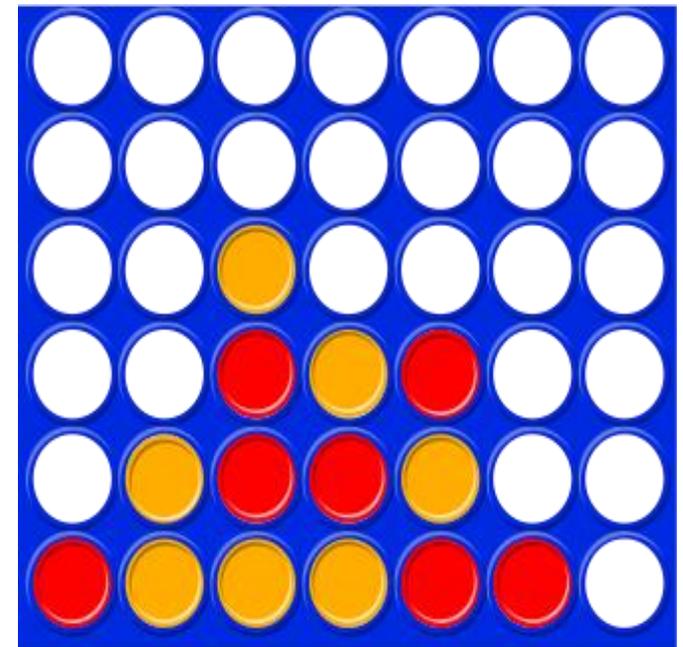


# Formaliser les besoins: les cas d'utilisation

Exercice: Donner le diagramme de cas d'utilisation du jeu de puissance 4

Il y a deux joueurs (jaune et rouge) qui jouent chacun à leur tour comme suit:

1. A chaque coup, ils peuvent mettre une de leurs pièces dans une des colonnes du tableau qui n'est pas encore pleine.
2. La pièce tombe dans la case la plus basse qui n'est pas encore occupée de la colonne.
3. Quand un joueur a aligné 4 de ses pièces, il a gagné.
4. Si aucun joueur n'y parvient et que le tableau est rempli la partie est nulle
5. En tout temps, un joueur peut abandonner la partie.



(source de l'image:

[https://fr.wikipedia.org/wiki/Puissance\\_4](https://fr.wikipedia.org/wiki/Puissance_4))

# Étude des besoins



**CONCLUSION**



**QUESTIONS ??**