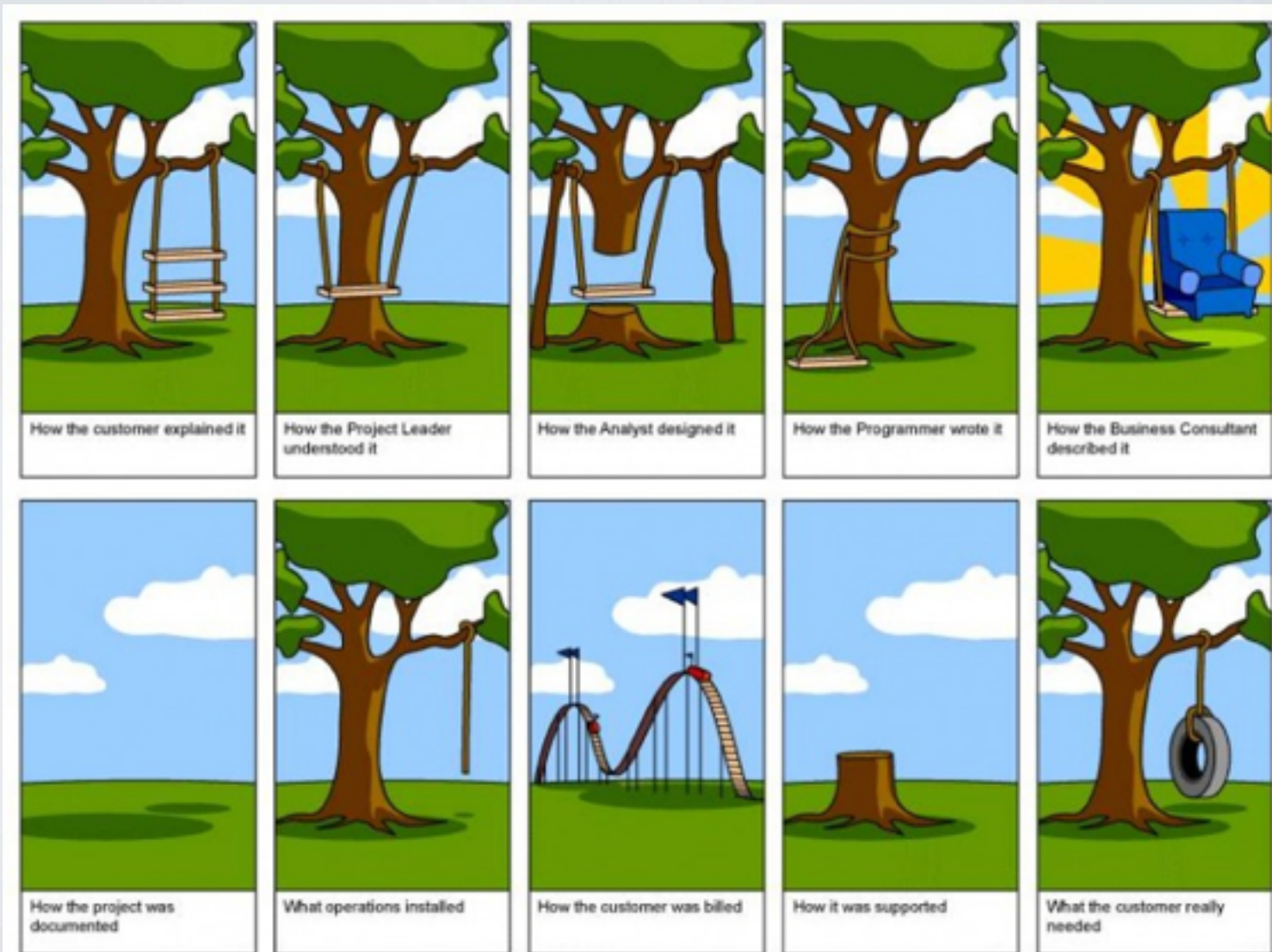
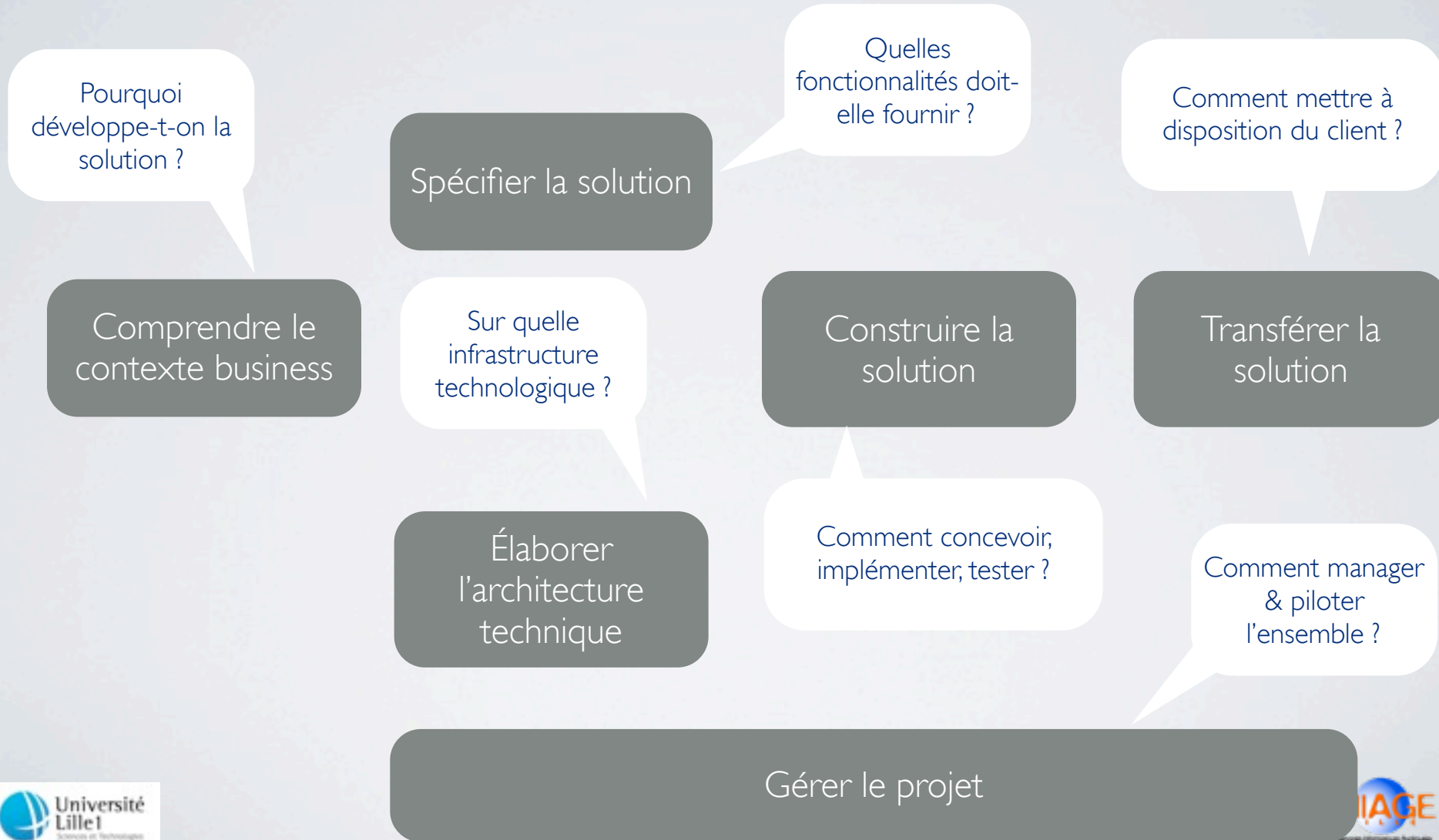


SPÉCIFICATION DE SOLUTION



LES GRANDES PRÉOCCUPATIONS D'UN PROJET DE DÉVELOPPEMENT



CONSTAT

- Il est nécessaire de :
 - formaliser ce que devra faire la solution
 - de définir précisément ce que devra faire la solution dans le cas d'un engagement forfaitaire
- Les besoins changent :
 - c'est dans la nature des choses
 - Il faut être capable d'intégrer ces changements et de les tracer

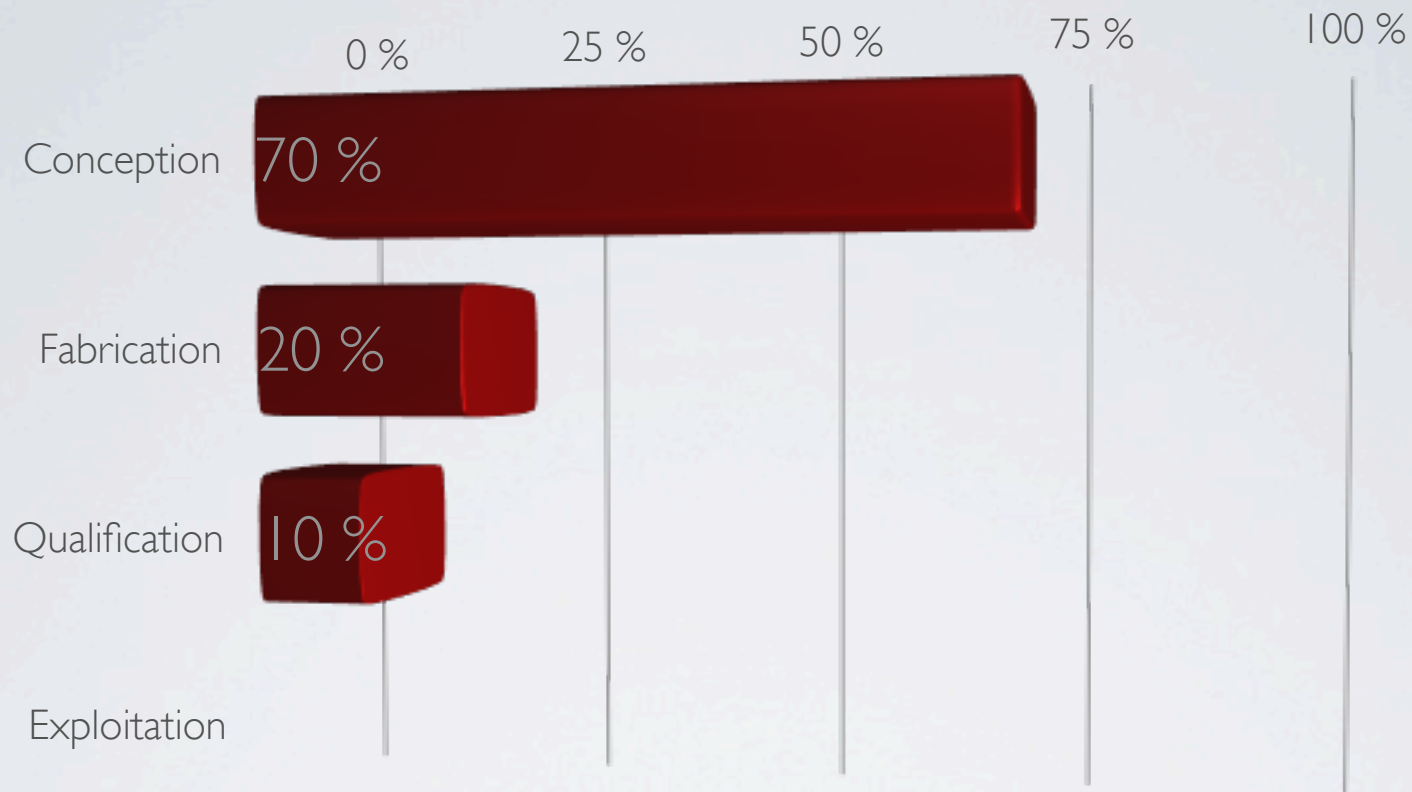
COMMENT FAIRE POUR SPÉCIFIER UNE SOLUTION ?

Activités ?

Rôles ?

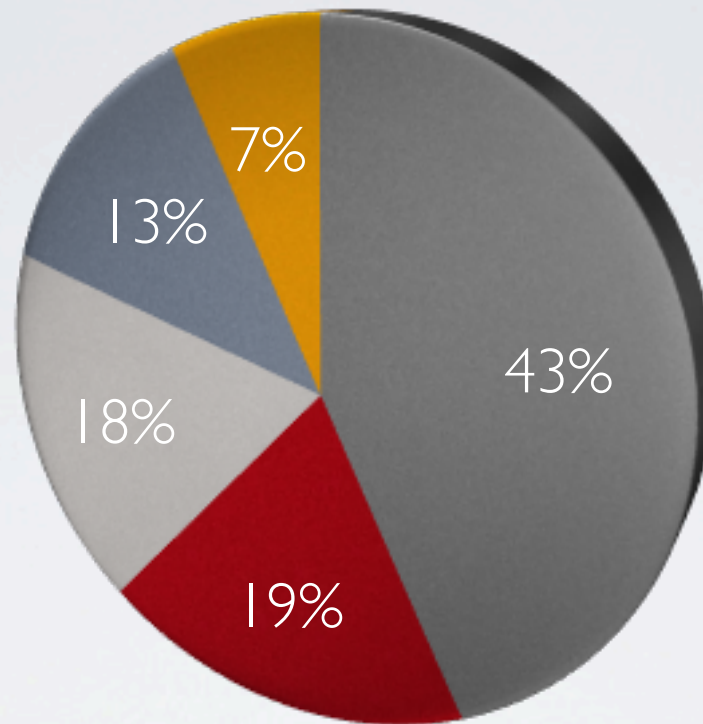
Techniques ?

Difficultés ?



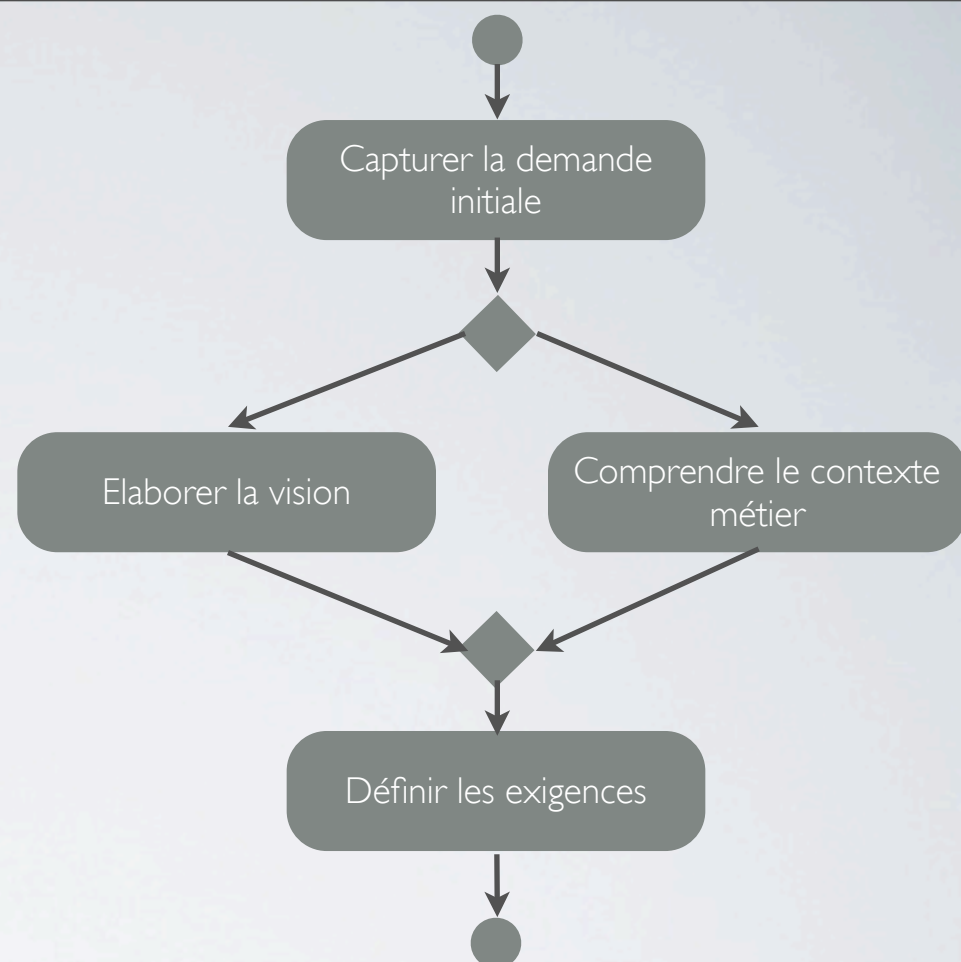
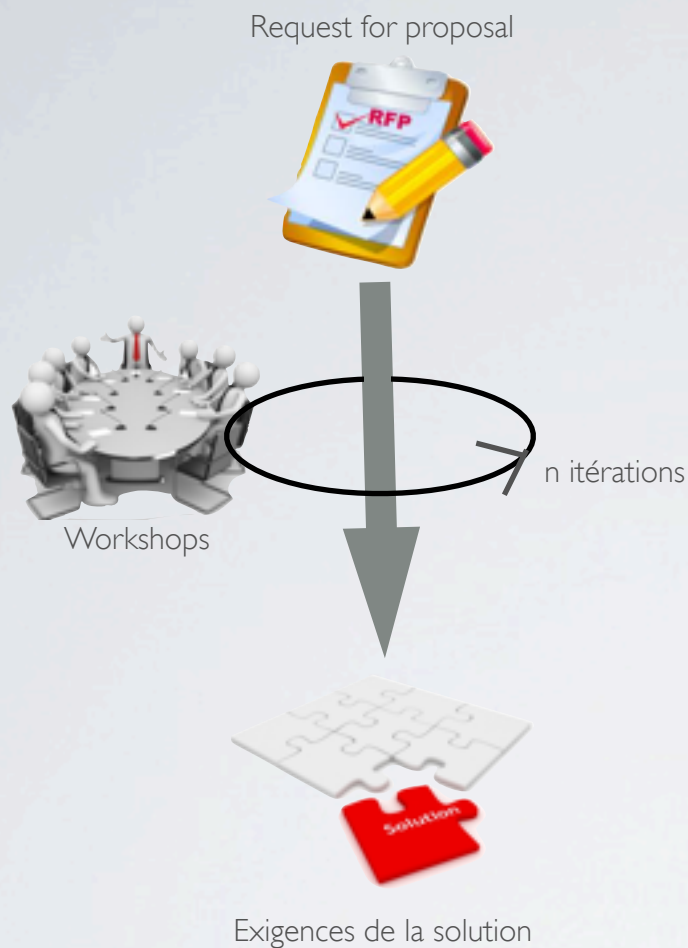
70% DES DÉFAUTS PROVIENNENT DE
L'EXPRESSION DES BESOINS ET LEURS
SPÉCIFICATIONS

UTILISATIONS DES FONCTIONS SPÉCIFIÉES



● Jamais ● Rarement ● Parfois ● Souvent ● Toujours

43% DES FONCTIONS SPÉCIFIÉES NE SONT JAMAIS UTILISÉES



SPÉCIFIER UNE SOLUTION

LES GRANDES PRÉOCCUPATIONS D'UN PROJET DE DÉVELOPPEMENT

Pourquoi
développe-t-on la
solution ?

**Comprendre
le contexte
business**

Spécifier la solution

Construire la
solution

Transférer la
solution

Élaborer
l'architecture
technique

Gérer le projet

Fonctionnalités I

Fonctionnalités 3

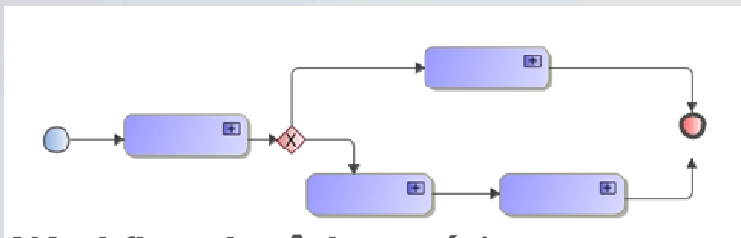
Fonctionnalités 2

Fonctionnalités 4



CONTEXTE BUSINESS

Comment sont-elles organisées ?



Workflow de tâches métier



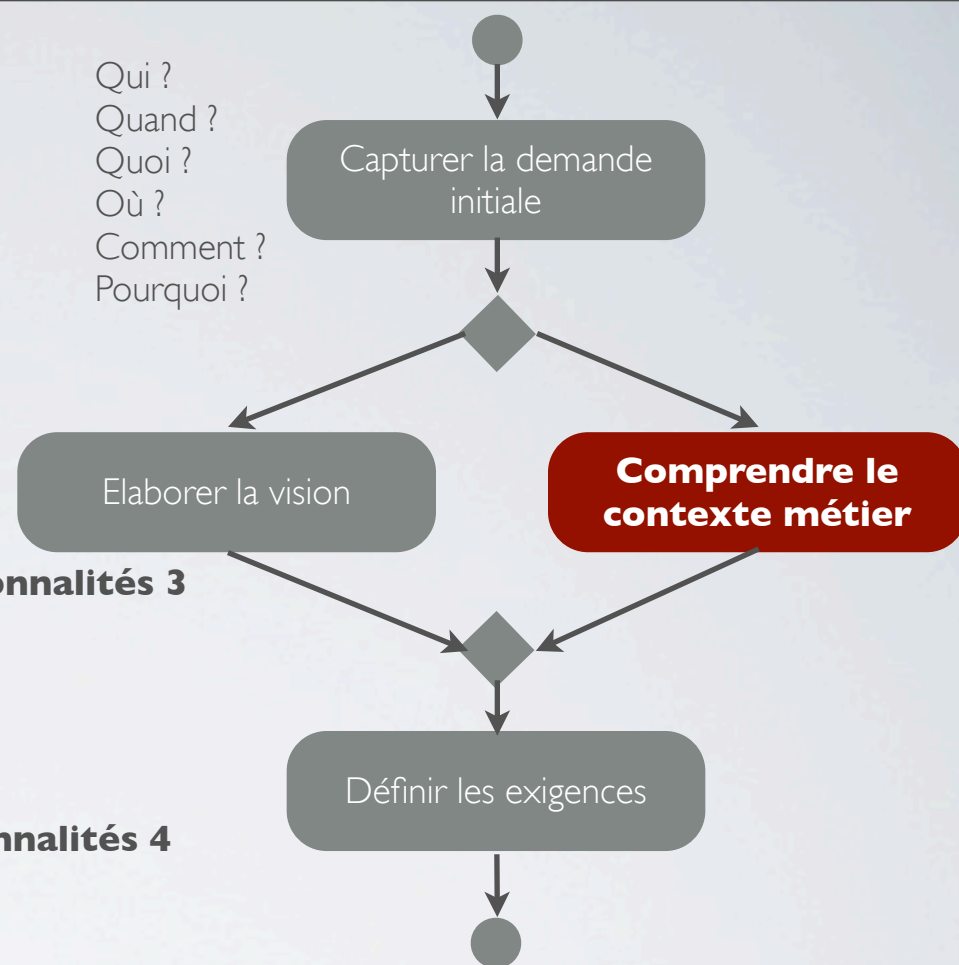
Fonctionnalités 1

Fonctionnalités 3

Fonctionnalités 2

Fonctionnalités 4

Qui ?
Quand ?
Quoi ?
Où ?
Comment ?
Pourquoi ?



CONTEXTE BUSINESS

COMPRENDRE ET PARTAGER LE CONTEXTE BUSINESS

- La RFP contient souvent une liste d'exigences non restituées dans leur contexte métier
 - Perte du lien entre les exigences et les besoins métiers qu'elles sont censées satisfaire
 - Pas de vision claire de l'enchaînement logique de la solution
- L'absence de définition précise du contexte métier de la solution est la source de problèmes qui se révéleront lors des phases ultérieures du projet ou lorsque les utilisateurs finaux utiliseront le système
 - Equipe de réalisation privée des moyens d'une réelle compréhension du problème
 - Priorisation métier des exigences difficile voire impossible
 - Incompréhensions ou interprétations divergentes pour les différentes parties prenantes clientes

EXIGENCES

PREMIÈRE DÉFINITIONS

- CMMI :
 - Condition ou capacité dont un utilisateur a besoin pour résoudre un problème ou atteindre un objectif.
 - Condition ou capacité que doit posséder un produit ou un composant de produit pour remplir un contrat, se conformer à une norme, une spécification ou tout autre document imposé formellement.
- Les exigences sont la spécification de ce qui va être implémenté. Elles correspondent à la description du comportement du système ou à des propriétés ou attributs du système.

Request for proposal



Exigences contractuelles



Workshops

n itérations



Exigences de la solution



EXIGENCES CONTRACTUELLES VS EXIGENCES DE LA SOLUTION

Request for proposal



Exigences contractuelles

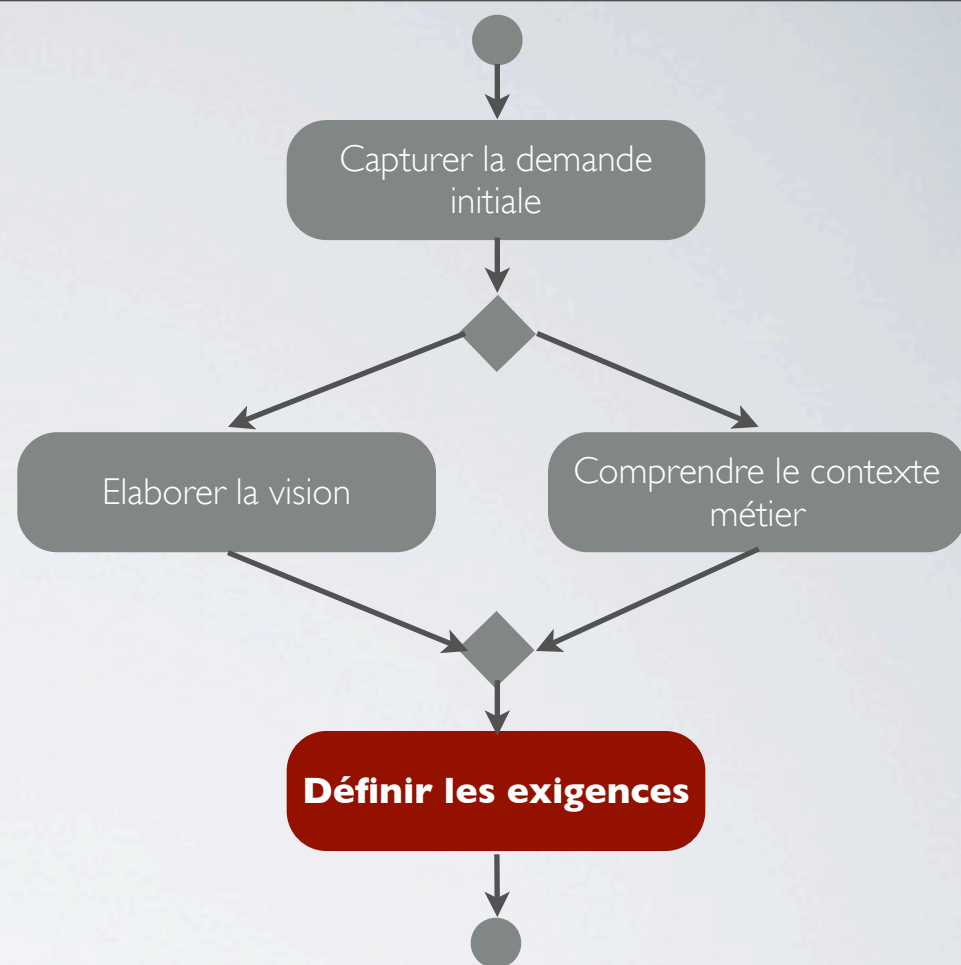


Workshops

/n itérations



Exigences de la solution



EXIGENCES CONTRACTUELLES VS EXIGENCES DE LA SOLUTION

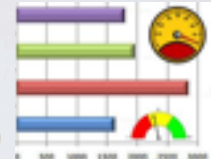


Exigences



Exigences fonctionnelles

Ce que doit faire la solution =
fonctionnalités fournies



...

Exigences non fonctionnelles

Contraintes sur les exigences
fonctionnelles

EXIGENCES FONCTIONNELLES ET NON FONCTIONNELLES



Le système doit calculer le taux d'endettement

Exigence Fonctionnelle

Taux d'endettement = Somme des charges / Sommes des revenus

Règle de gestion

Le temps de réponse ne doit pas dépasser n secondes.

Exigence non fonctionnelle

Le résultat du calcul est affiché en rouge, en gras sur 4 caractères

Règle d'ergonomie

EXIGENCES FONCTIONNELLES ET NON FONCTIONNELLES

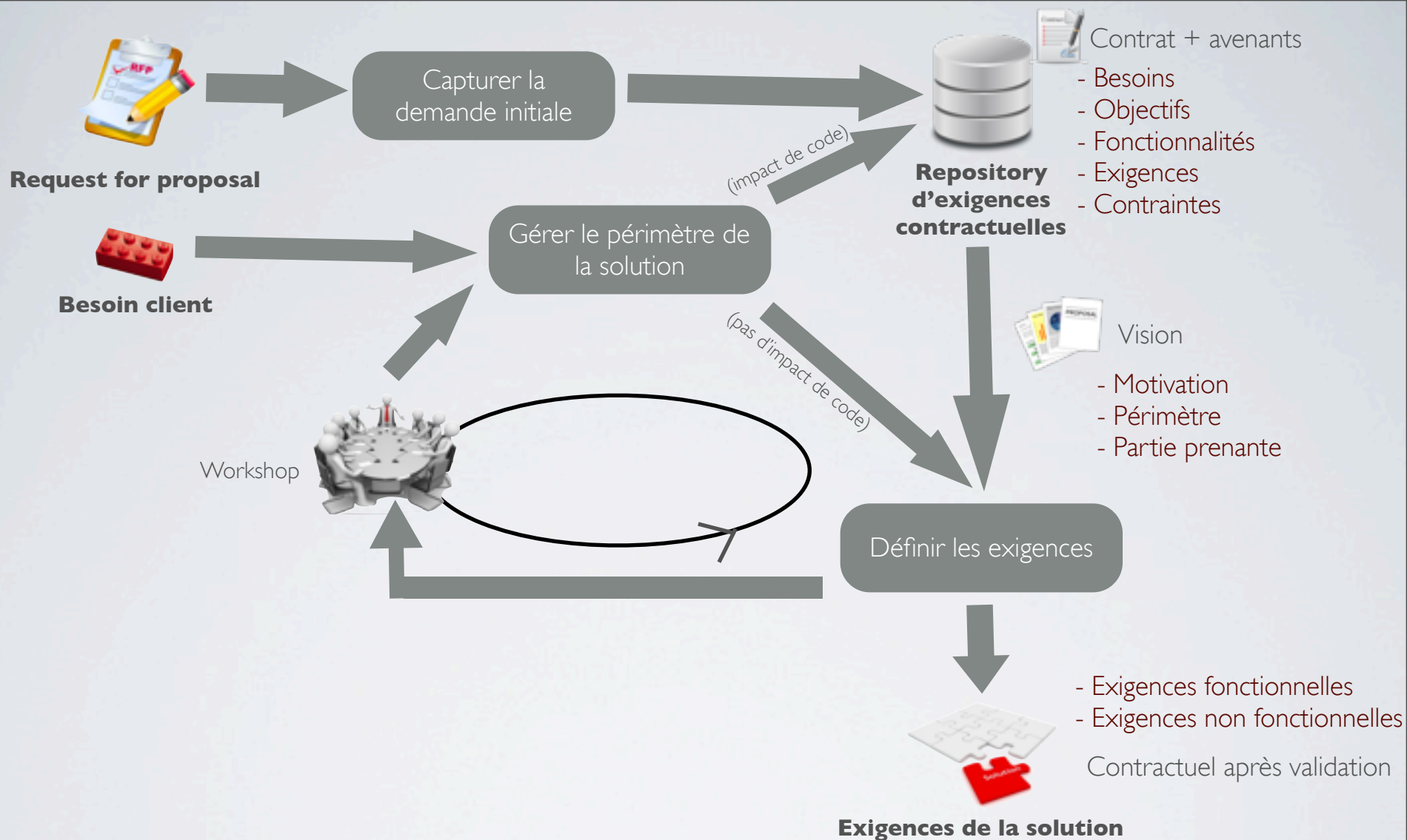
DE LA COLLECTE D'EXIGENCES ...

- Exigence :
 - **Unique** : adresse | seul sujet, pas de doublon
 - **Testable** : peut se vérifier
 - **Complète** : permet d'assurer les travaux de réalisation
 - **Faisable** : réalisable en vue des contraintes projet
 - **Non ambiguë** : sa description doit être claire & précise
 - **Cohérente** : s'il y a contradiction avec les exigences contractuelles, elle doit être validée par le client

QU'EST CE QUE LA GESTION D'EXIGENCES ?

... À LA GESTION D'EXIGENCES, ...

- Créer un référentiel des exigences contractuelles (demandes client)
- Spécification des exigences & élaboration d'un référentiel des exigences de la solution
 - Typier les exigences (techniques, fonctionnelle, business, ...)
 - Gérer la « granularité »
- Mettre en place une matrice de traçabilité entre exigence contractuelles et exigences solution
- Gérer les mises à jours



MODÈLE DE GESTION D'EXIGENCES

EXIGENCE CMMI REQM (NIVEAU 2)

- Purpose : manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products.
- Specific Practices by Goal
 - SG 1 Manage Requirements
 - SP 1.1 Obtain an Understanding of Requirements
 - SP 1.2 Obtain Commitment to Requirements
 - SP 1.3 Manage Requirements Changes
 - SP 1.4 Maintain Bidirectional Traceability of Requirements
 - SP 1.5 Identify Inconsistencies Between Project Work and Requirements

... PUIS AU DÉVELOPPEMENT D'EXIGENCES

- Développer une culture exigences à tous les niveaux (métier, MOA, MOE)
- S'assurer de la qualité des exigences (analyses, brainstorming, ...)
- Avoir une approche « composant »
- Outiller la gestion d'exigence

EXIGENCE CMMI RD (NIVEAU 3)

- Purpose : produce and analyze customer, product, and product component requirements.
- Specific Practices by Goal
 - SG 1 Develop Customer Requirements
 - SP 1.1 Elicit Needs
 - SP 1.2 Develop the Customer Requirements
 - SG 2 Develop Product Requirements
 - SP 2.1 Establish Product and Product Component Requirements
 - SP 2.2 Allocate Product Component Requirements
 - SP 2.3 Identify Interface Requirements
 - SG 3 Analyze and Validate Requirements
 - SP 3.1 Establish Operational Concepts and Scenarios
 - SP 3.2 Establish a Definition of Required Functionality
 - SP 3.3 Analyze Requirements
 - SP 3.4 Analyze Requirements to Achieve Balance
 - SP 3.5 Validate Requirements

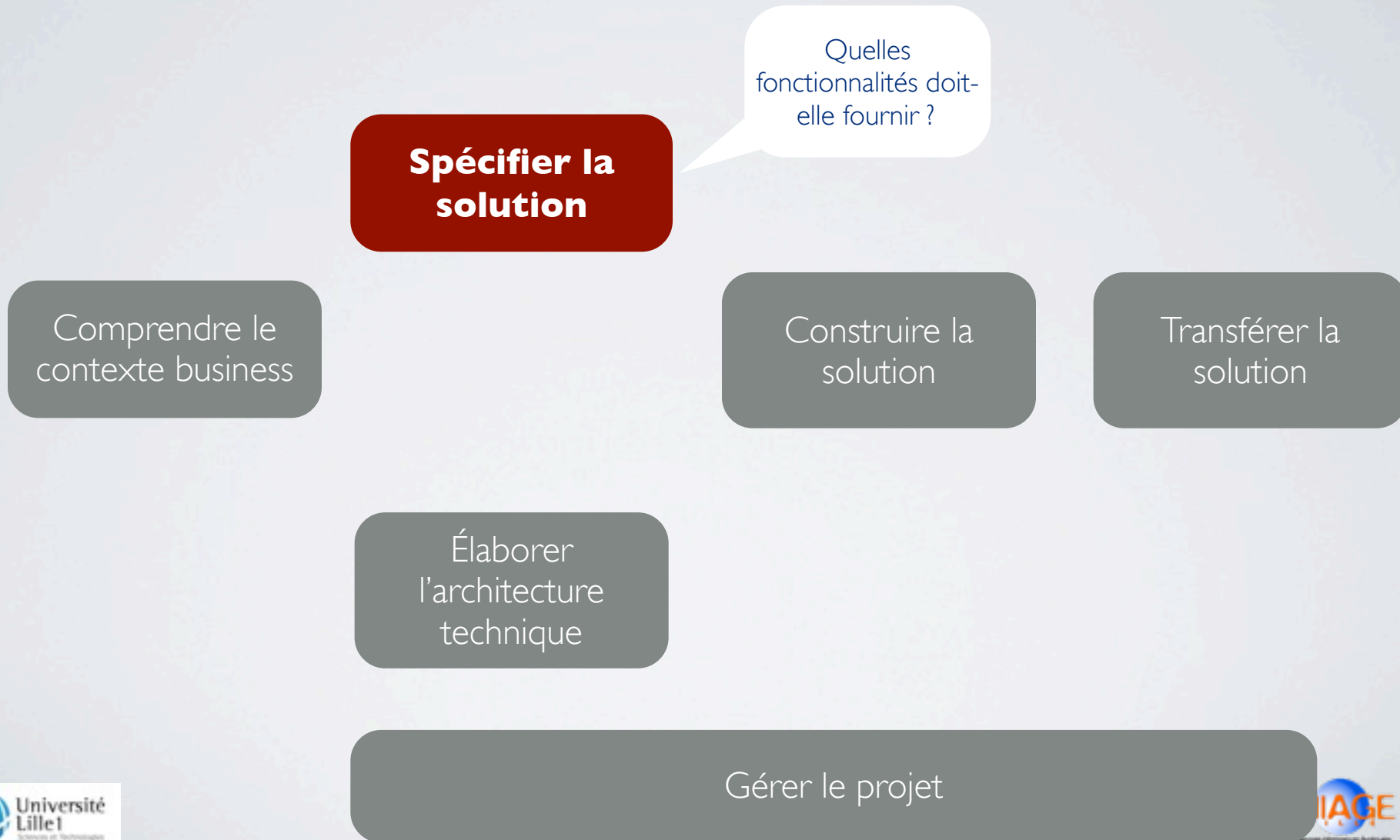
A QUOI ÇA SERT ?

- Assurer la couverture d'un cadrage :
on définit le projet à partir des exigences contractuelles
- Assurer la couverture des spécifications :
on définit la solution à partir des exigences solutions et en s'assurant que toutes les exigences contractuelles sont couvertes
- Assurer la couverture d'un plan de test :
de la même façon que la traçabilité *exigences contractuelles* \Leftrightarrow *exigences solutions*, on peut s'assurer de la couverture des plan de tests par le lien *exigences solutions* \Leftrightarrow *cas de tests*

EXEMPLES D'OUTILS

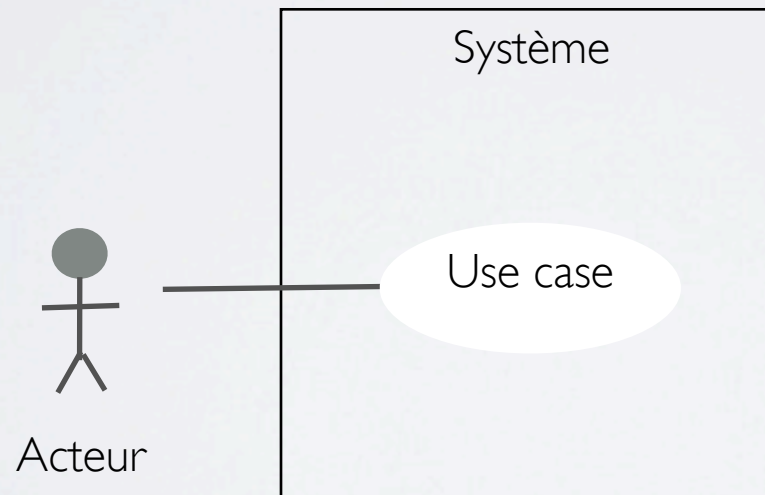
- Reqtify : extraction d'exigence, base, lien avec autres outils
- HP Quality Center / IBM Rational : base d'exigences, liens tests
- un tracker + un workflow

LES GRANDES PRÉOCCUPATIONS D'UN PROJET DE DÉVELOPPEMENT



TECHNIQUES DE SPÉCIFICATION

- Blabla
- Uses cases



POURQUOI LES UC ?

Techniques 'Traditionnelles'

On définit les fonctions que devra procurer le système :

Exemple :

REQ-17: le système devra permettre la recherche des réservations par nom et prénom du Client

REQ-25: le système devra permettre la vérification des disponibilités si le client n'a pas de réservation

REQ-37 : le système doit afficher une liste de réservations avec identité client, date début, durée du séjour

REQ-38 : le système doit afficher le détail d'une réservation avec

Techniques de Story (Use cases, User stories)

On raconte, sous forme d'histoire, comment l'acteur utilise le système :

Exemple :

1. Le réceptionniste sélectionne la recherche par identité et renseigne le nom (obligatoire) et le prénom (facultatif)
2. Le système recherche les réservations correspondant à l'identité renseignée et en affiche la liste en indiquant le nom, le prénom, la date de début, la durée du séjour et le nombre de nuitées
3. Le réceptionniste sélectionne la réservation du client

POURQUOI LES UC ?

Exigences 'Traditionnelles'

PERTE DU LIEN AVEC LE MÉTIER

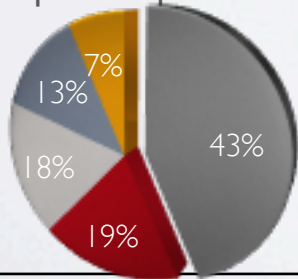
- Problème d'alignement des applications sur les besoins métier

PAS D'ENCHAÎNEMENT DES FONCTIONNALITÉS

- Les fonctionnalités sont à plat et leur ordre d'exécution n'est pas explicite
- Difficulté à les prioriser

VUE INTERNE DU SYSTÈME

- Le système doit faire ...
- Fonctionnalités pas ou peu utilisées



Use case

LIEN EXPLICITE AVEC LE MÉTIER

- Core UC = tâche informatisée

EXPLICITE L'ENCHAÎNEMENT DES FONCTIONNALITÉS

- Toutes les fonctionnalités nécessaires à la réalisation de l'objectif d'un utilisateur

PRIVILÉGIE LA VUE UTILISATEUR

- Raconte sous forme d'histoire comment l'utilisateur utilise le système pour atteindre son objectif
- Limite le piège des fonctionnalités non utilisées (validation des utilisateurs)

USER STORY

En tant que **Réceptionniste**, je veux **Rechercher une réservation par identité client** pour **ouvrir un séjour**



1. Le réceptionniste choisit la recherche par identité et renseigne le nom (obligatoire) et le prénom (facultatif)
2. Le système recherche les réservations correspondant à l'identité renseignée et en affiche la liste en indiquant le nom, le prénom, la date de début, la durée du séjour et le nombre de nuitées
3. Le réceptionniste choisit la réservation du client.
4. Le système affiche le détail de la réservation (en plus des informations précédemment affichées) : nombre de personnes, nombre de petit déjeuners, prix par nuitée, prix du petit déjeuner, montant total.

USE CASE / USER STORY

Use case

TECHNIQUE ÉPROUVÉE & MATURE

CORE UC = TÂCHE MÉTIER INFORMATISÉE

Exemple : ouvrir un séjour

Des plus

- Lien explicite avec le métier
- Regroupe toutes les fonctionnalités nécessaire à la réalisation d'une tâches métier

Des moins

- Difficile de trouver le bon niveau de granularité
- Trop gros pour être développé en 1 itération (cas de méthode itératives)

User story

TRÈS POPULAIRE DANS LE MONDE AGILE

USER STORY = FONCTIONNALITÉ COMPLÈTE POUR L'UTILISATEUR

Exemple : En tant que Réceptionniste, je veux Rechercher une réservation par identité client pour ouvrir un séjour

Des plus

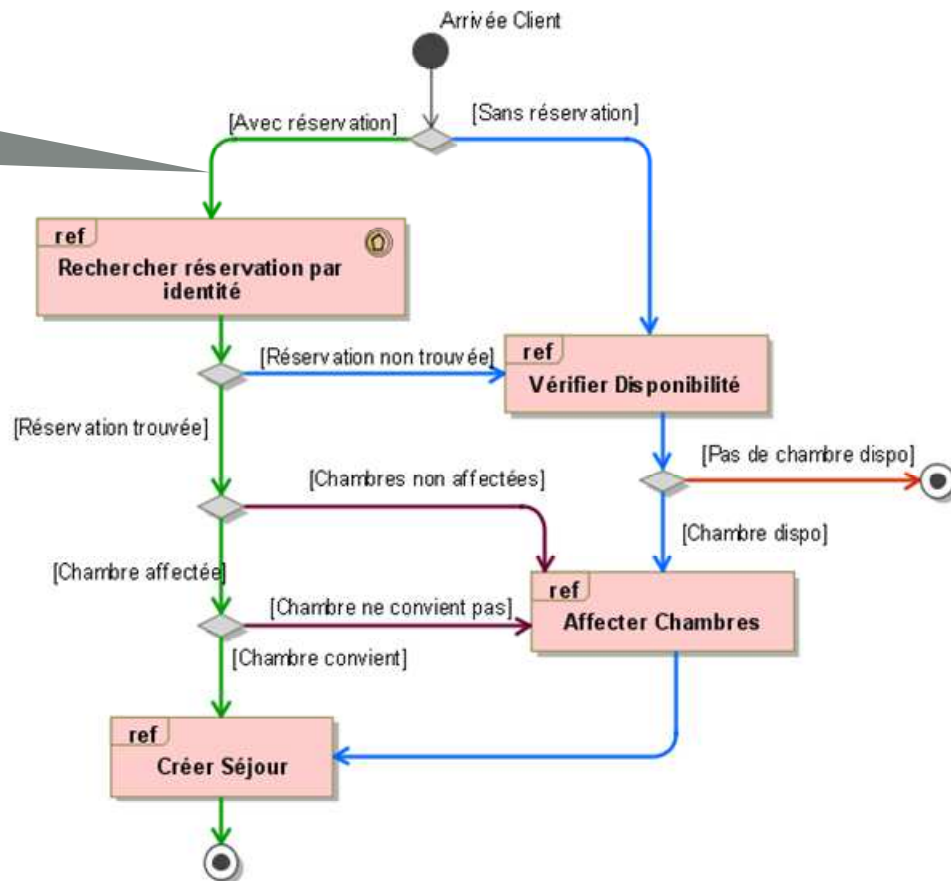
- Facile à identifier & spécifier
- Bon niveau de backlog pour 1 itération (cas de méthode itératives)

Des moins

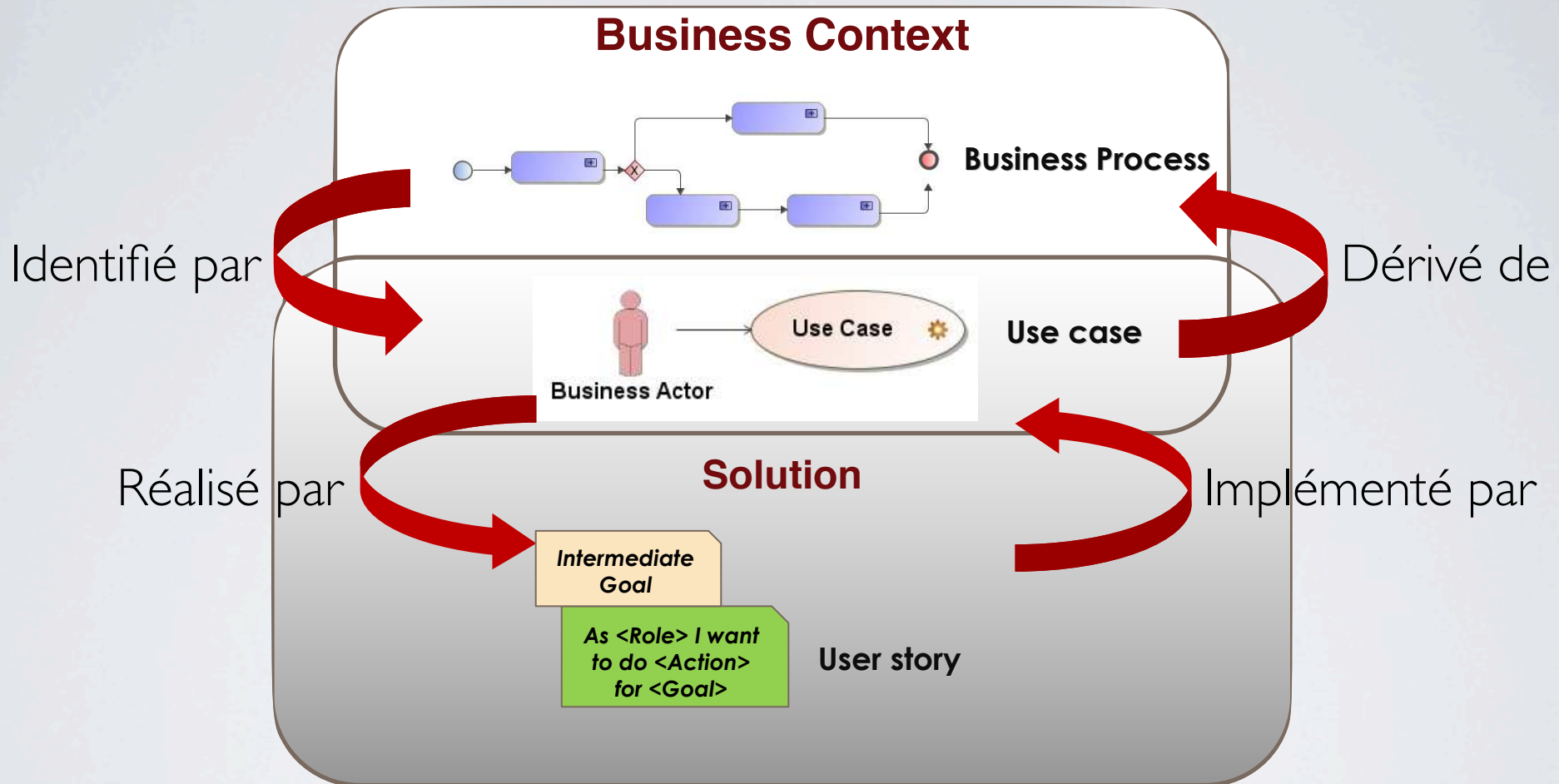
- Risque de liste de fonctionnalités à plat
- Pas de lien explicite avec le métier
- Enchainement non formalisé

WORKFLOW D'UC : INTÉGRER LE MEILLEUR DES 2

User
Story



SPÉCIFIER UNE SOLUTION



Les use cases sont le pivot entre le métier et les exigences