



METHODES AGILES De gestion de projet

Septembre - Octobre 2012

- Contexte Agile
- **XP** (eXtreme Programming)
- **SCRUM**
- L'ingénierie du logiciel
- Pilotage par enjeux et risques

SOMMAIRE

Contexte Agile

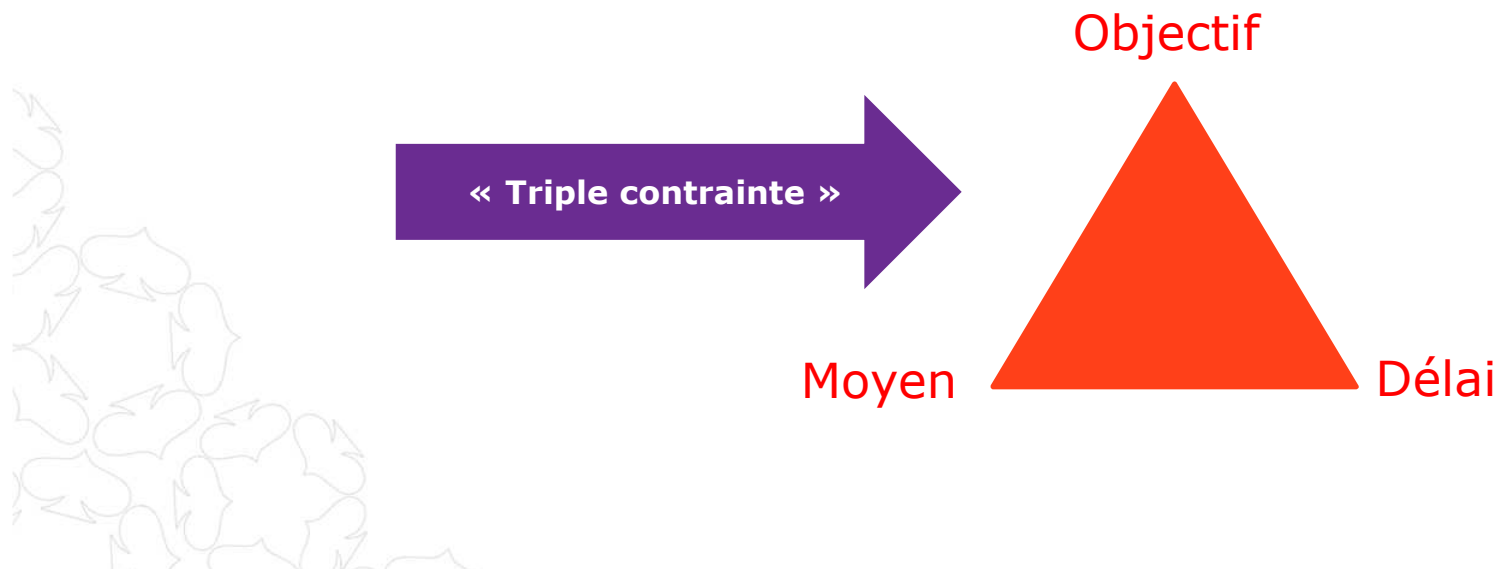
- Agiles et gestion de projet
- Pourquoi les méthodes Agiles
- Contexte des méthodes Agiles
- Panorama
- Principes de base

Agiles et gestion de projet

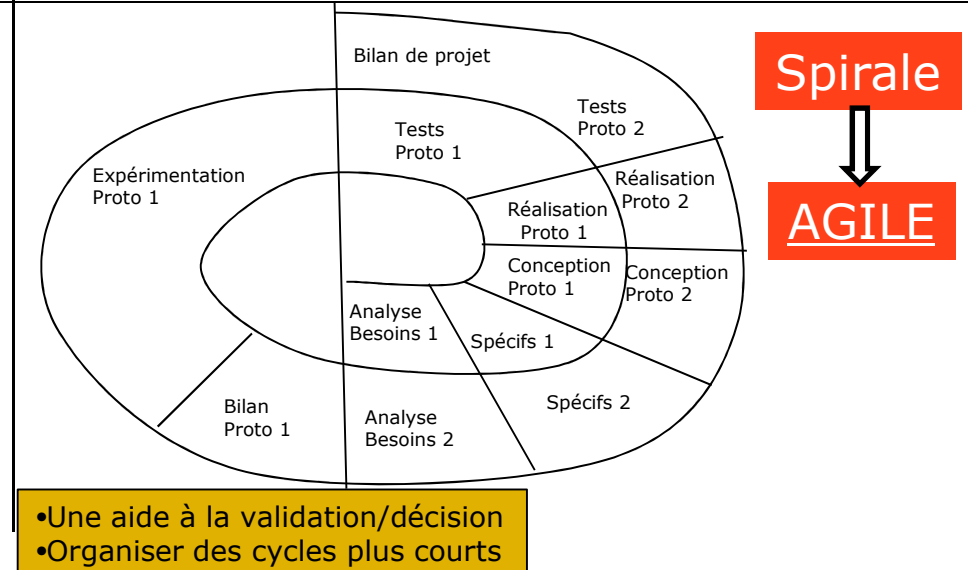
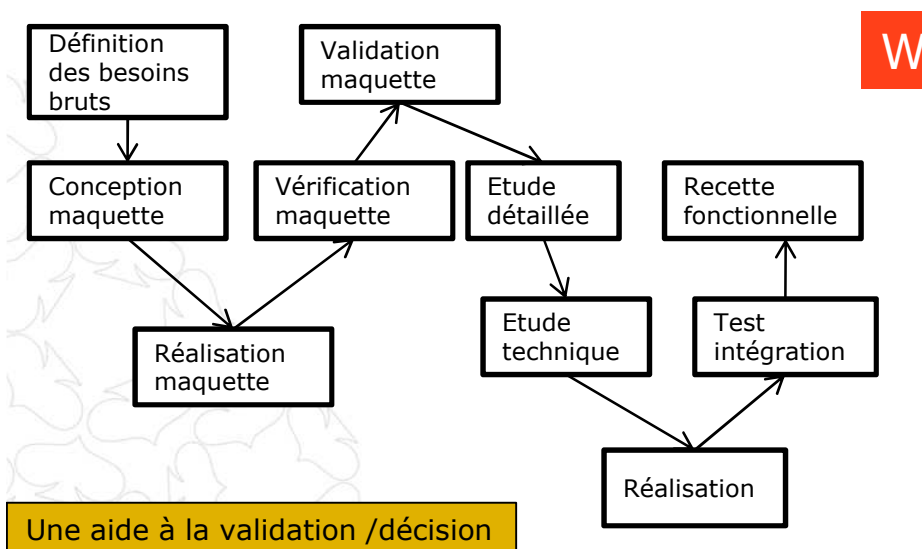
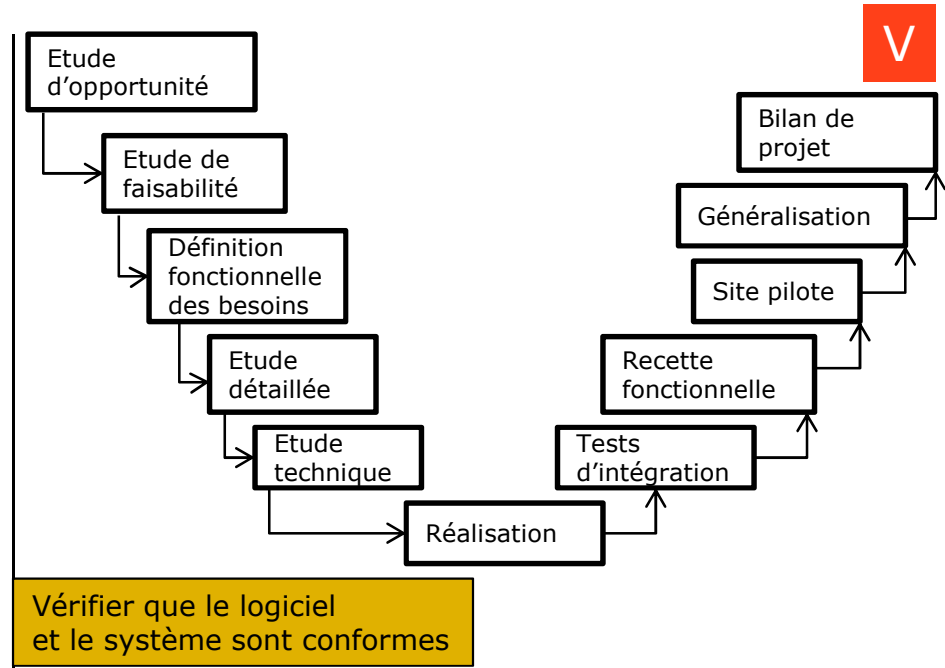
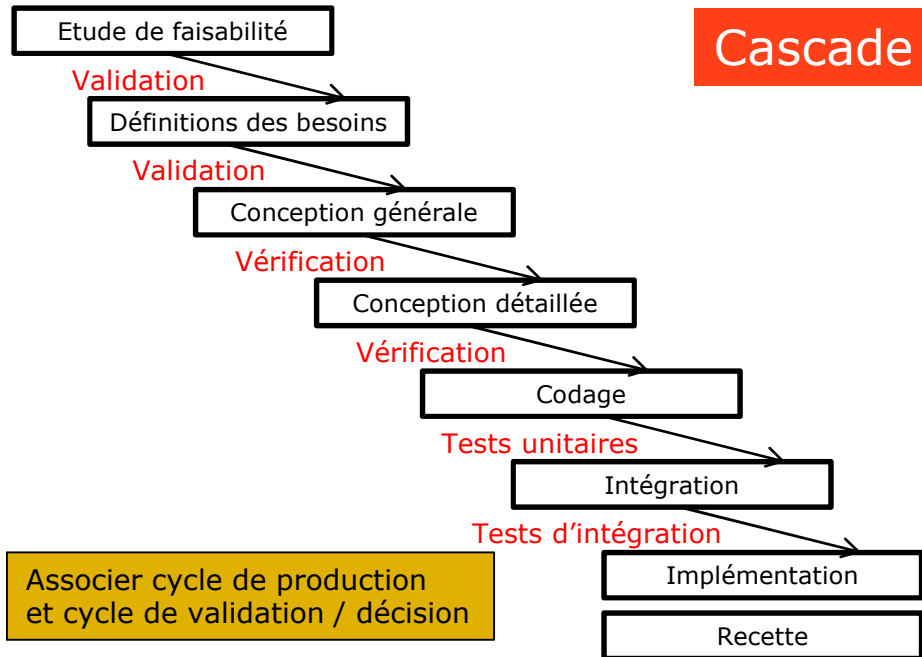
- **Projet :**

Processus unique, qui consiste en un ensemble d'activités coordonnées et maîtrisées comportant des dates de début et des dates de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques telles que les contraintes de délais, de coûts et de ressources

(Norme ISO 10006 : 2003)

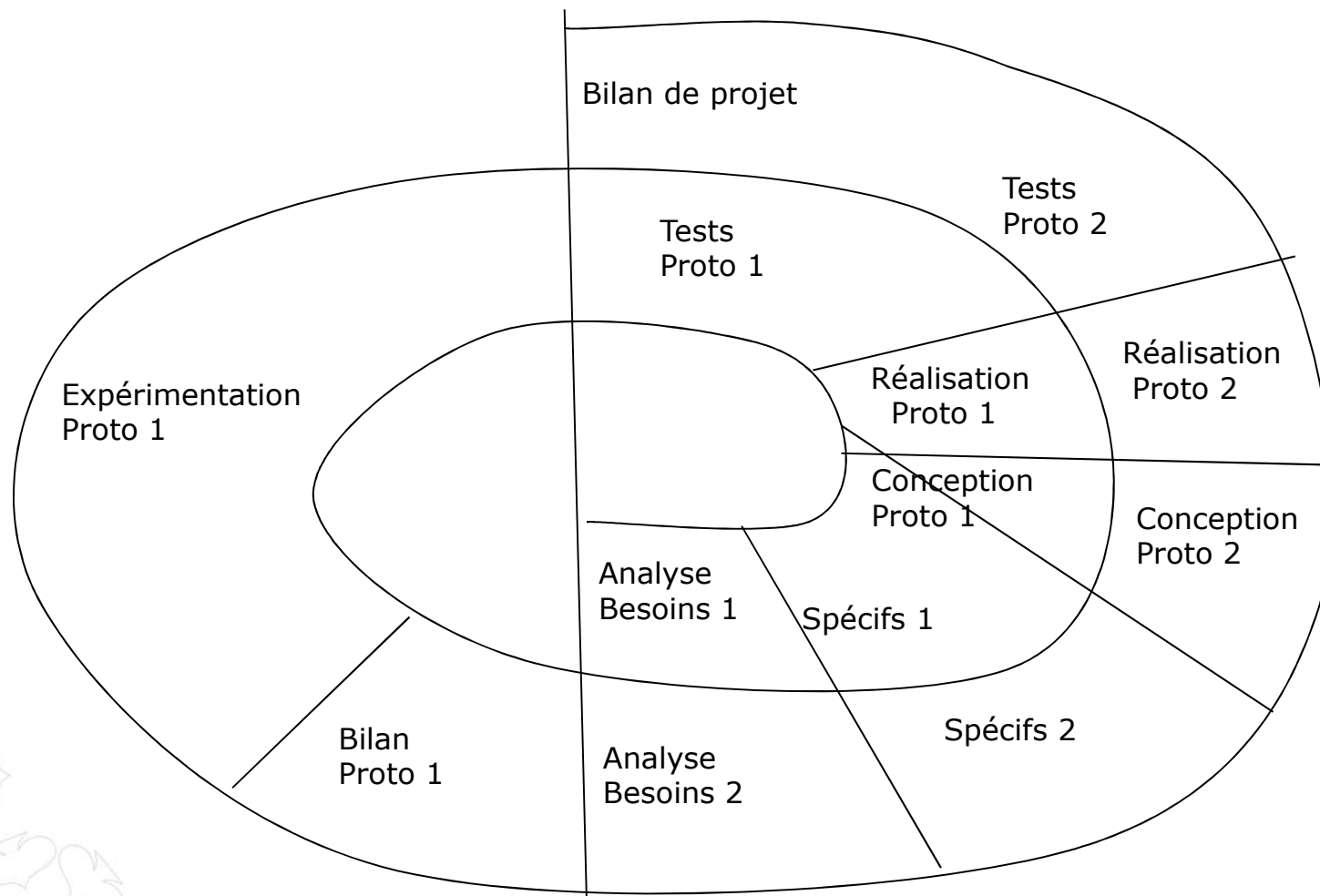


Contexte Agile Agiles et cycles



Contexte Agile

Agiles et cycles



Pourquoi des méthodes agiles

- **Raccourcir les cycles de développement**
- **Mieux intégrer le client**
- **Organiser l'expression des besoins**
- **Rendre le travail de développement plus facile**
- **Réagir aux méthodes lourdes**

Contexte des méthodes agiles

- **Pas une méthode universelle**
- **Projet simple**
- **Environnement non-critique**
- **Equipé motivée**
- **Client impliqué**
- **Objectif : Satisfaction client immédiate**
- **Supporter les imprévus**

Contexte Agile

PANORAMA



▪ RAD

- Rapid Application Development
- 1991

▪ XP

- eXtreme Programming
- Kent Beck
- 2000

▪ SCRUM

- Scrum = mêlée
- Mike Beedle
- Accent sur les pratiques de planification

▪ Crystal

- Alistair Cockburn
- Une famille de méthodes
- Amélioration continue

▪ ASD

- Adaptive Software Development
- Jim Highsmith
- Issu d'une observation de projets+

▪ FDD

- Feature Driven Development
- Feff De Luca et Peter Coad

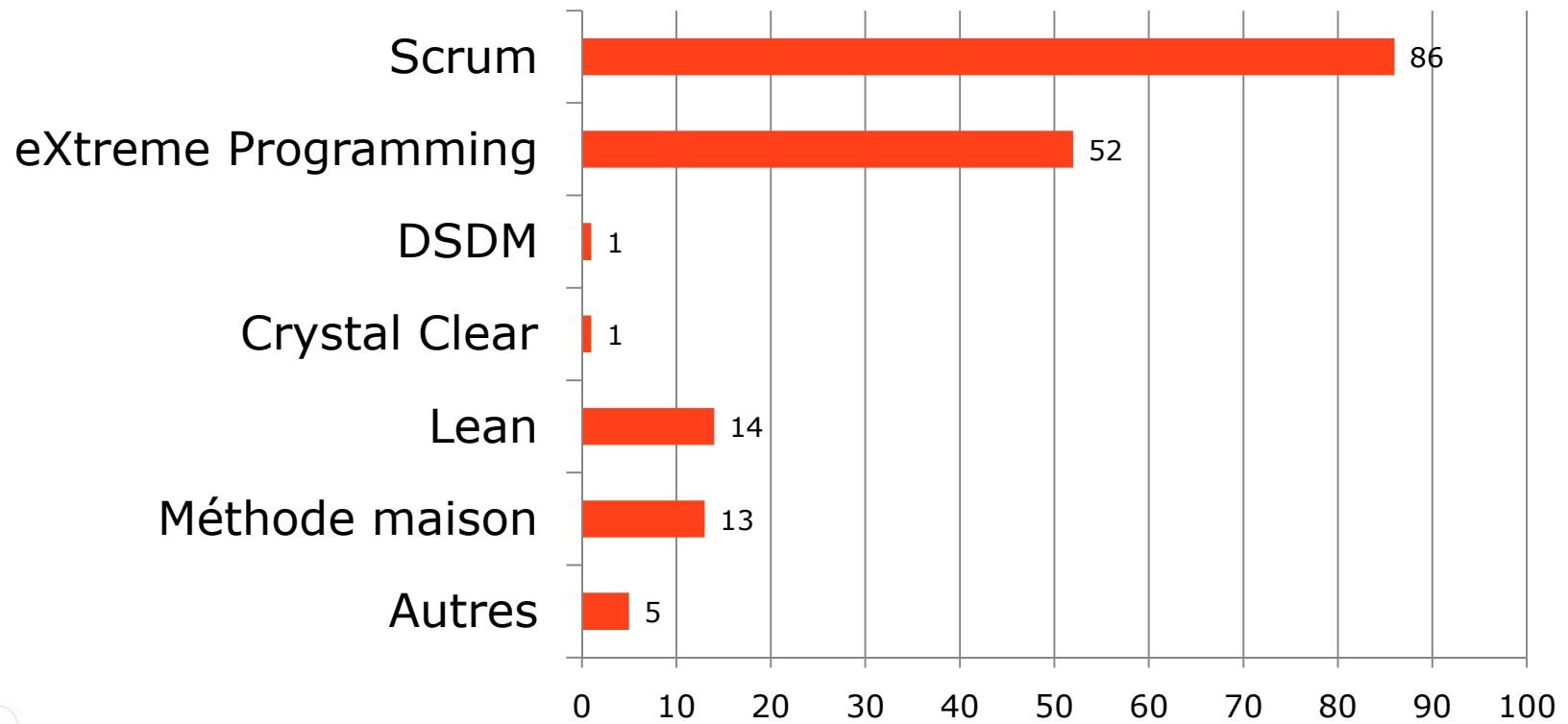
▪ DSDM

- Dynamic System Development Methode
- Consortium d'entreprises
- Forte inspiration RAD

▪ LEAN

- Trouve ses sources au Japon
- Sa forme la plus élaborée est aujourd'hui chez Toyota

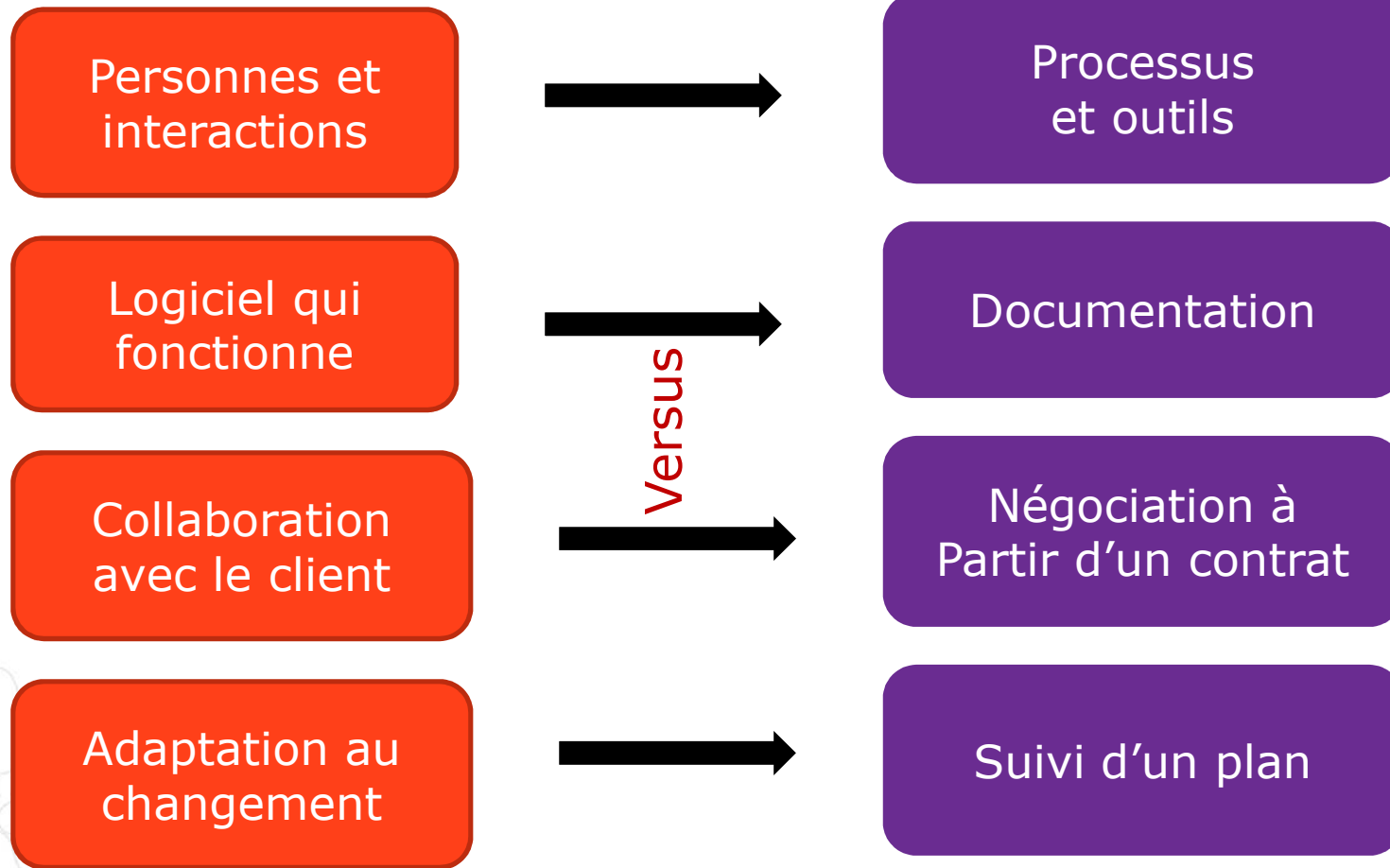
Utilisateur de méthodes Agiles en %



- Etude de SUG (Scrum User Group)
- Réalisée en 2009
- Disponible sur www.frenchsug.org

Contexte Agile Principes de base

Issues du manifeste pour le Développement logiciel Agile
www.agilemanifest.org



- **Spécification et validation permanente des exigences.**
- **Participation de l'utilisateur final aux groupes de travail.**
- **Groupe de travail disposant du pouvoir de décision.**
- **Autonomie et organisation centralisée de l'équipe (motivation).**
- **Niveau méthodologique variable en fonction des enjeux du projet.**
- **Pilotage par les enjeux et les risques.**
- **Recherche d'excellence technique de la conception**
- **Vision graphique d'une modélisation nécessaire et suffisante.**
- **Vision de la documentation nécessaire et suffisante.**
- **Planification stratégique globale basée sur les itérations rapides.**
- **Réalisation en jalons par prototypage actif itératif et incrémental.**
- **Normes et techniques raisonnables de qualité du code.**
- **Architecture à base de composants, gestion des changements.**
- **Recherche continue d'optimisation et d'amélioration des pratiques.**

SOMMAIRE

eXtreme Programming

- Présentation
- L'équipe
- La gestion de projet
- Les autres pratiques

eXtreme Programming Présentation



- **1996** : Projet Chrysler : Kent Beck – Ron Jeffries
- **2000** : Parution de Extreme Programming Explained de K. Bech
- **XP** : Un processus projet léger permettant à une équipe de se centrer sur la réalisation d'une solution
- **XP** : S'adresse à des petites équipes qui souhaitent réaliser rapidement des logiciels et réagir rapidement au changement
- **XP** : Un ensemble de pratiques qui couvrent les activités de réalisation d'un logiciel – programmation, planification, management des parties prenantes.

eXtreme Programming

Les pratiques de XP

- **Les pratiques de gestion de projet :**

- **Livraisons fréquentes**
- **Planification itérative**
- **Client sur site**
- **Rythme durable**

- **Les pratiques de programmation :**

- **Conception simple**
- **Remaniement**
- **Développement piloté par des tests**

- **Les pratiques de gestion de projet :**

- **Programmation en binôme**
- **Responsabilité collective du code**
- **Règles de codage**
- **Métaphore**
- **Intégration continue**

eXtreme Programming

Les valeurs de XP

La communication par une meilleure visibilité

- S'accorder sur une vision commune
- Synchroniser les actions individuelles
- Communication surtout orale, directe
- Communication écrite : Code, test, recette

Le feedback comme outil de réduction du risque

- Pouvoir corriger le tir fréquemment

La simplicité comme garantie de productivité

- Pas ou peu de généricité
- Pas ou peu de + non demandé

Le courage de prendre les bonnes décisions

- Principe de communication transparente

eXtreme Programming

L'équipe

- **Client**
- **Testeur**
- **Programmeur**
- **Manager**
- **Coach**
- **Tracker**



Client

- **Un membre à part entière de l'équipe**
- **Son rôle :**
 - Spécifie les fonctionnalités à implémenter
 - Assure les tests fonctionnels

Testeur

- **Un programmeur de l'équipe**
- **Son rôle :**
 - Implémenter les tests de recettes spécifiés par le client

Programmeur

- **Un analyste – Concepteur Un membre à part entière de l'équipe**
- **Son rôle :**
 - Analyse, conçoit et écrit l'application
 - Analyse, conçoit et exécute les tests unitaires
 - Estime la charge des scénarios à implémenter

Manager

- **Un responsable hiérarchique des développeurs et/ou du client**
- **Son rôle :**
 - Assure la responsabilité de l'infrastructure dans laquelle travaille l'équipe

Coach

- **Un manager, un technicien crédible, un expert XP**
- **Son rôle :**
 - Manager l'équipe
 - Piloter le projet
 - Communiquer



Tracker

- **Un manager, un technicien crédible, un acteur neutre**
- **Son rôle :**
 - Contrôler l'avancement des tâches
 - Détecter les dérives et corriger

L'équipe – Combinaisons de rôles

- Plusieurs rôles pour une même personne

	Programmeur	client	Testeur	Tracker	Manager	Coach
Programmeur		X	~	~	X	~
client			V	X	X	X
Testeur	~	V		X	X	X
Tracker	~	X	X		~	~
Manager		X	X	~		X
Coach	~	X	X	~	X	

Combinaisons :

V : Bonnes

X : Mauvaises

~ : Envisageables mais non sans risques

- Plusieurs personnes pour un même rôle (Programmeur, tracker, client)

eXtreme Programming

La gestion de projet

- **Les pratiques de gestion de projet**
 - **Client sur site**
 - **Rythme durable**
 - **Planification itérative**
 - **Livraisons fréquentes**

- **Les phases**
 - **La phase initiale**
 - **Les phases d'une itération de livraison**
 - **Les phases d'une itération de développement**

La gestion de projet – Le client sur site

- **Le client est intégré à l'équipe projet**
- **Le client définit les besoins**
- **Le client arbitre les priorités**
- **Le client participe quotidiennement au projet**
- **Le client apporte ses compétences métier**
- **Le client définit les tests fonctionnels**



La gestion de projet – Rythme durable

- **Des estimations de charges pertinentes**
 - Se méfier du penchant à la sous estimation des techniciens

- **Une planification réaliste**

Pas ou peu d'heures supplémentaires

- **Anticiper les aléas**

- **En les planifiant**
- **En les provisionnant**

- **Lisser la charge**

Eviter les périodes

- **Prendre en compte régulièrement les changements de contexte**

La gestion de projet – Planification itérative

- **Cycle de livraison**

- Des nouvelles fonctionnalités visibles par le client

Itération initiale



Itération de livraison 2



Itération de livraison 3



Itération de livraison n

- Itérative



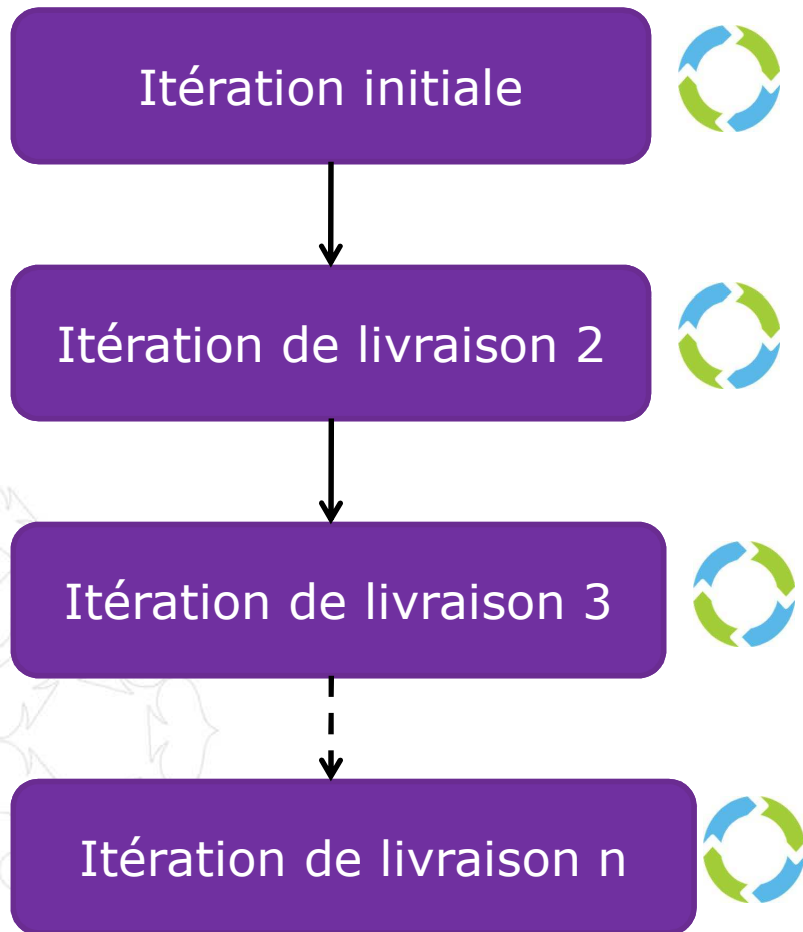
- Itérative et **incrémentale**



La gestion de projet – Planification itérative

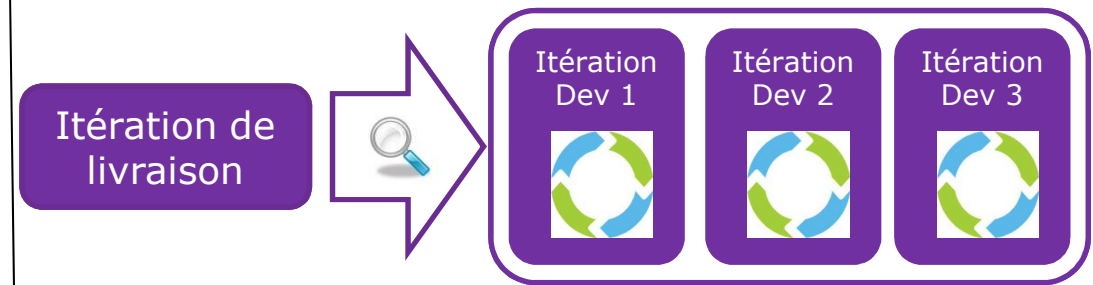
▪ Cycle de livraison

- Des nouvelles fonctionnalités visibles par le client



▪ Cycle de développement

- Des nouvelles tâches réalisées par les développeurs

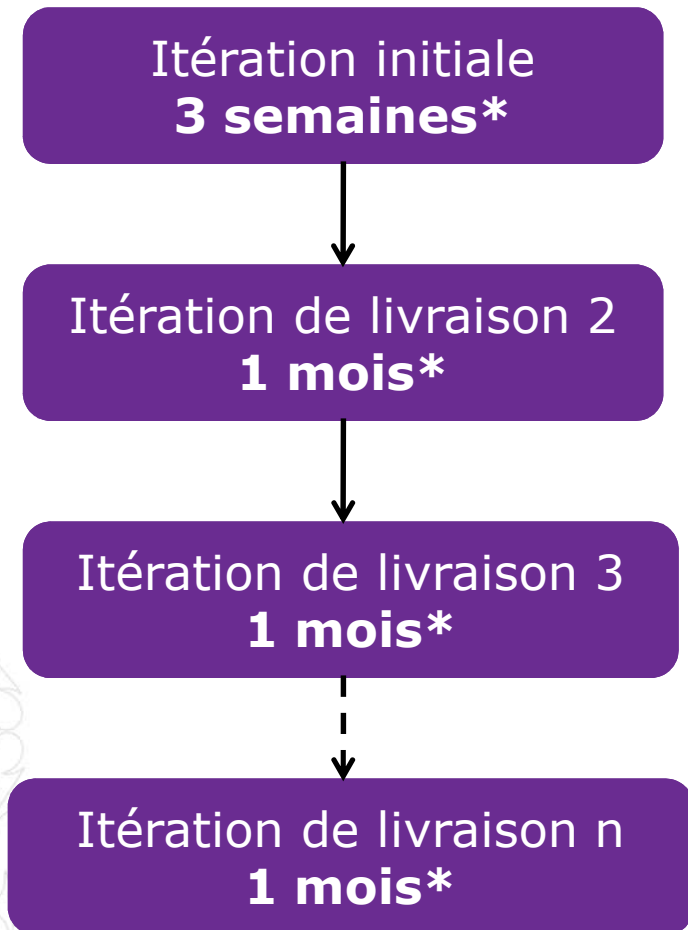


▪ Pour chaque cycle



- Explorer
- Engager
- (Construire) Piloter

La gestion de projet – Livraison fréquentes



*:exemple

- **Feedback pour le client**

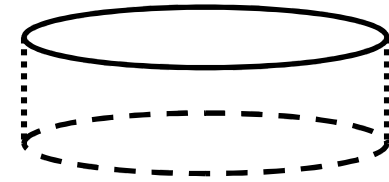
- Information sur l'avancement du projet
- Organisation de l'expression du besoin

- **Feedback pour l'équipe**

- Cristallisation de l'avancement
- Retour d'expérience en environnement réel

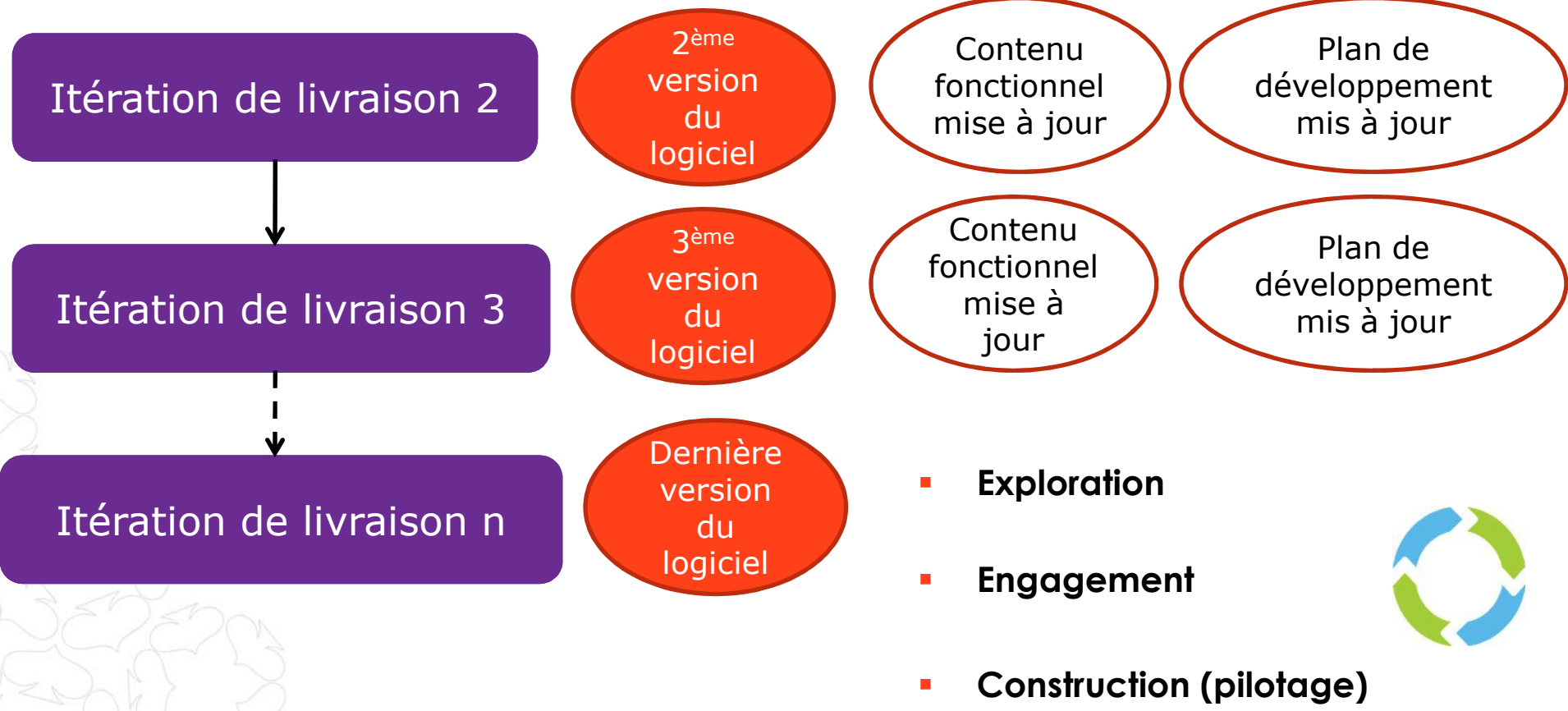
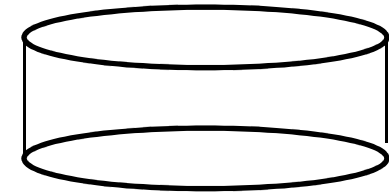
« Il n'existe pas de fréquence idéale mais dans la pratique cette fréquence est souvent égale ou inférieure à un mois ».

Freddy Mallet



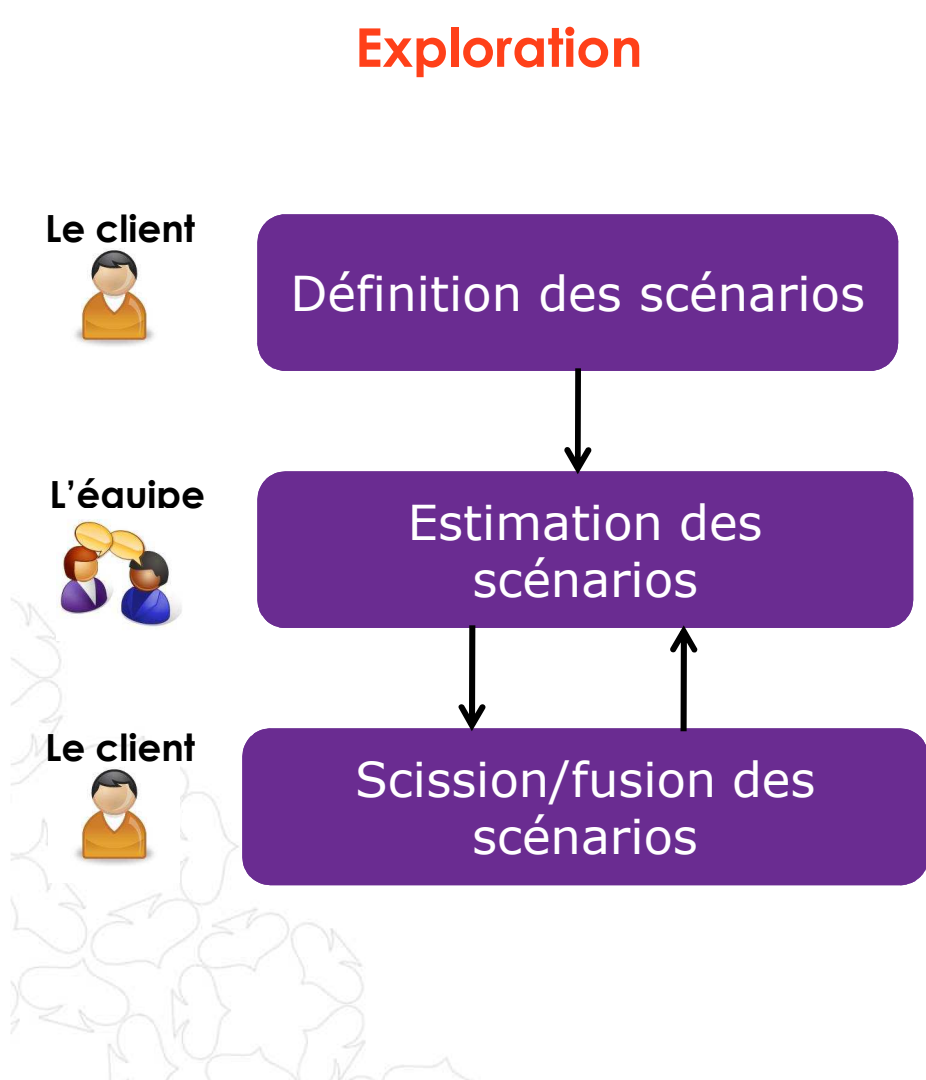
- **Le choix d'une démarche Agile est faite**
- **Construire l'équipe projet**
- **Définir en superficie quasi exhaustive le périmètre fonctionnel**
- **Elaborer (éventuellement) un macro modèle des données**
- **Recenser les contraintes et les choix techniques**
- **Evaluer une charge et un coût global**
- **Construire un macro planning**
- **Valider l'approche Agile**

La gestion de projet – La phase initiale

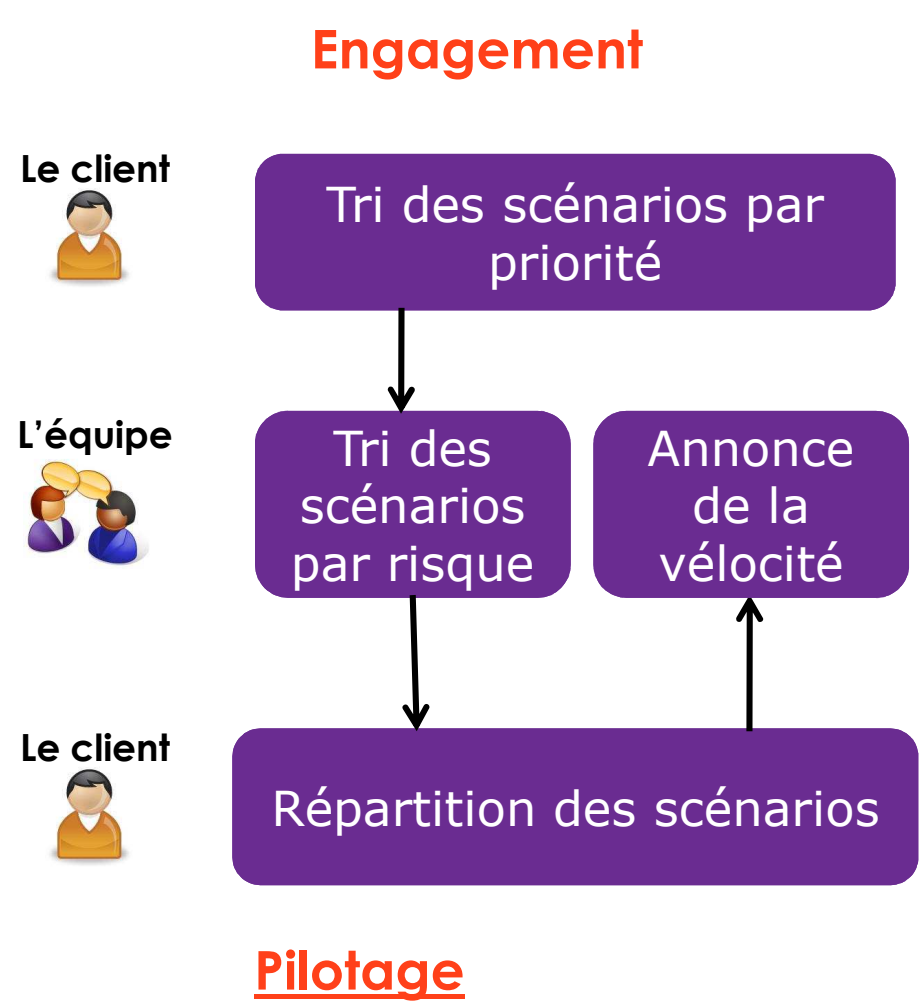


La gestion de projet – Les phases d'une livraison

Exploration



Engagement



Pilotage

La gestion de projet – Phases d'exploitation

Définition des scénarios

Exploration

Le client



Définition des scénarios

L'équipe



Estimation des scénarios

Le client



Scission/fusion des scénarios

■ Scénario :

Un comportement du futur système

- Qui peut être estimé
- Qui peut être testé
- Qui peut être décrit sur une fiche

Exemple :
Maxi une semaine

■ Scénario :

- Une fonction
- Une contrainte
- Une règle
- ...

■ Description :

- **User story**
- Cas d'utilisation
- Procédure de test
- ...

La gestion de projet – Phases d'exploitation

User story -1

Exploration

Le client



Définition des scénarios



L'équipe



Estimation des scénarios



Le client



Scission/fusion des scénarios

- Les scénarios sont décrits sur des fiches format A5

Fiche N°	Titre :
Priorité fonctionnelle :	Risque technique : Charge :
Description :	

User Stories:

- Fiche N°** : Permet d'identifier le scénario de façon unique
- Titre** : Résumé du scénario (1 phrase)
- Priorité** fonctionnelle : 1 (indispensable) / 2 (essentiel) / 3 (utile)
- Risque** technique : Fort / Moyen / Faible
- Charge** : Estimation en points
- Description** : Quelques lignes (1 à 5 phrases) de la fonctionnalité souhaitée

La gestion de projet – Phases d’exploitation User story -2

→ Exploration

Le client



Définition des scénarios

L'équipe



Estimation des scénarios

Le client



Scission/fusion des scénarios

Des exemples

Fiche N° : 12 **Titre : Saisie coordonnées**
Priorité fonctionnelle : **1** Risque technique : Charge :

Description : Saisie des coordonnées du client
(Liste des coordonnées)

Fiche N° : 6 **Titre : Connexion**
Priorité fonctionnelle : **1** Risque technique : Charge :

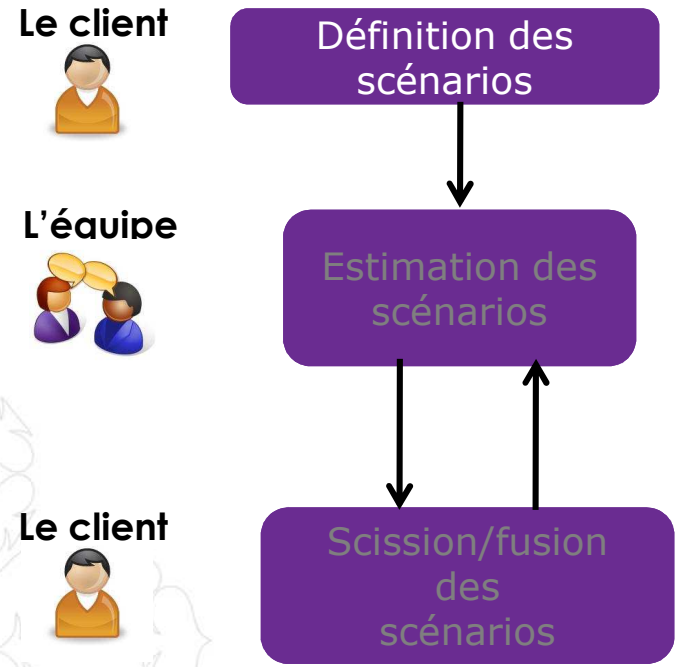
Description : Au démarrage demander le nom et le mot de passe

Fiche N° : 7 **Titre : Contrôle autorisation**
Priorité fonctionnelle : **1** Risque technique : Charge :

Description : Vérifier si le nom et le mot de passe sont enregistrés

La gestion de projet – Phases d’exploitation User story -3

➔ Exploration



▪ Des exemples

Fiche N° : 32	Titre : Liste des préférences
Priorité fonctionnelle : 1 Risque technique : Charge :	
Description : Le client doit pouvoir conserver une liste personnelle de cours et les organiser par filières	

Les scénarios peuvent aussi décrire des exigences

Fiche N° : 25	Titre : Temps de lancement
Priorité fonctionnelle : 1 Risque technique : Charge :	
Description : Le temps de lancement de l’application doit être inférieur à 5 secondes sur un poste d’exploitation	

La gestion de projet – Phases d'exploitation

Estimation des scénarios

 **Exploration**

Le client



Définition des scénarios

L'équipe



Estimation des scénarios

Le client



Scission/fusion des scénarios

Fiche N° :	Titre :
Priorité fonctionnelle : 1	Risque technique : Charge :

- Une séance de travail entre le client et l'équipe
- Des questions réponses pour clarifier les fiches scénarios
- Une évaluation de la charge par analogie à des précédents scénarios (difficulté des premiers scénarios)

Se baser la première fois sur une expression calendaire (1 point = 1 semaine de tps idéal)
- Charge de développement, de test, de recette
- Une évaluation exprimée en valeur relative : Le point (Cf. Vitesse dans phase d'engagement)
- Eventualité de « break » pour « carottages » (essai, prototype...)
- Préalable souhaitable : Définir l'architecture technique (base de composants fonctionnelles, techniques...)

La gestion de projet – Phases d'exploitation Scission / Fusion des scénarios

Exploration

Le client



Définition des scénarios

L'équipe



Estimation des scénarios

Le client



Scission/fusion des scénarios

Granularité trop grosse

➤ Charge trop importante

➔ Scission

➤ Estimation moins fiable

Granularité trop fine

➤ Charge de « coordination »

➔ Fusion

Fiche N° : 6

Titre : Connexion

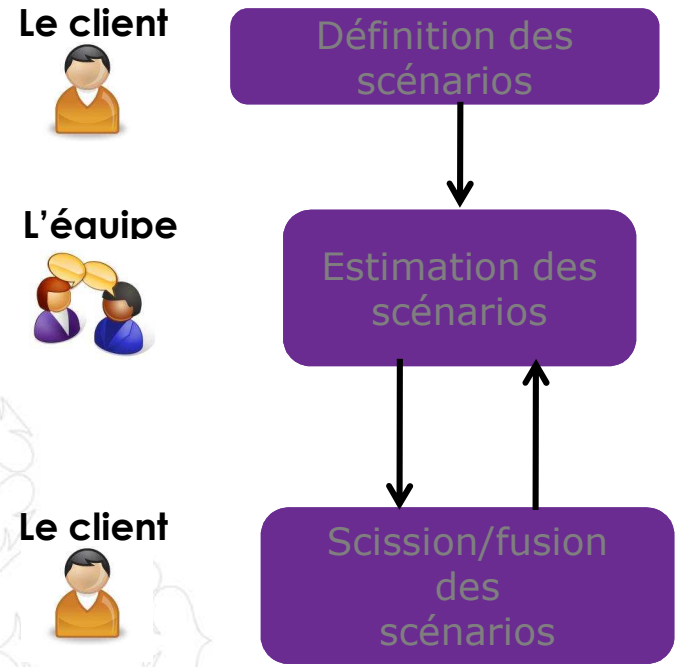
Priorité fonctionnelle : 1 Risque technique : Charge :

Description : Au démarrage demander le nom et le mot de passe

La gestion de projet – Phases d’engagement

Durée d’une phase d’exploitation

Exploration

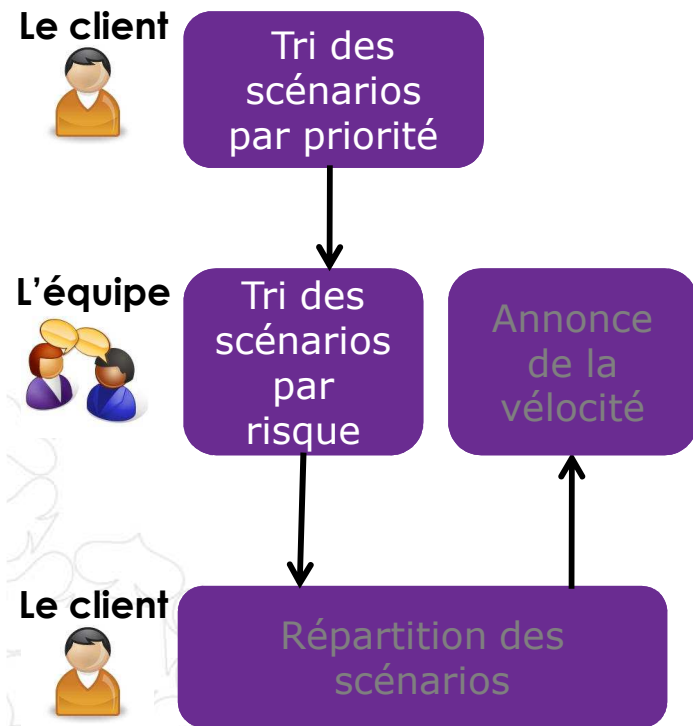


- **Début du projet** → **Peut durer plusieurs jours**
- **En cours du projet** → **Diminue pour durer en moyenne quelques heures**

Au début du projet → La phase d'exploration peut être confondue avec la 1^{ère} itération

La gestion de projet – Phases d’engagement Tri des scénarios

Engagement

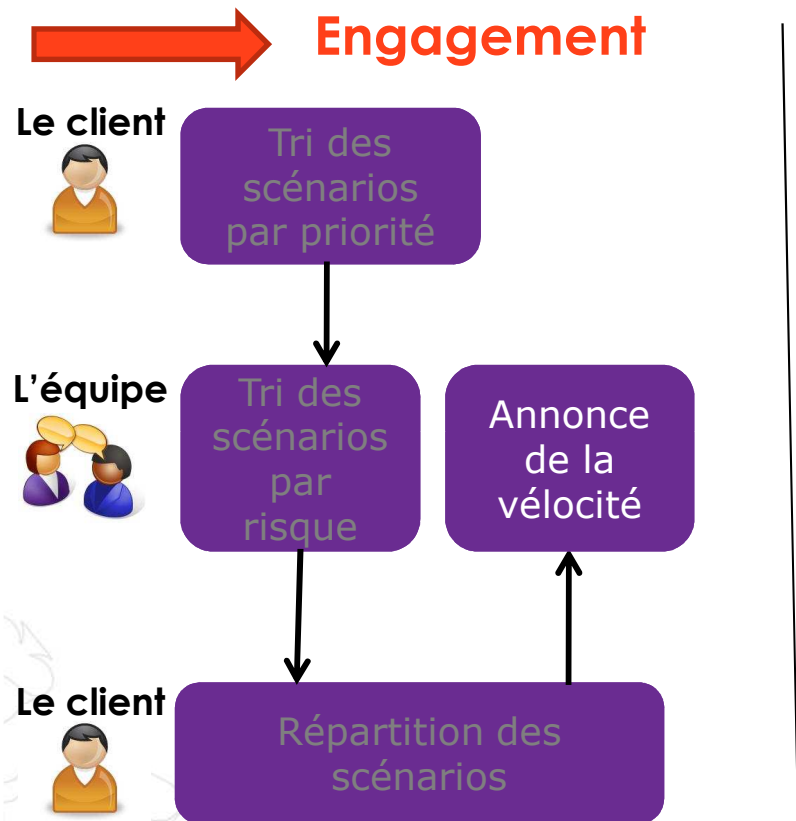


Classement des scénarios:

- Par priorité fonctionnelle (par le client)
- Par risques techniques (par l’équipe)

		Risque		
		Faible	Moyen	Fort
Priorité	Indispensable			
	Essentiel			
	Utile			

La gestion de projet – Phases d’engagement Annonce de la vélocité



Vélocité de l'équipe

- Total des estimations en points des scénarios qui ont été totalement et correctement implémentés au cours de l'itération précédente
- Problème de la 1^{ère} itération
- Nombre de points de scénarios client que l'équipe peut traiter en une itération

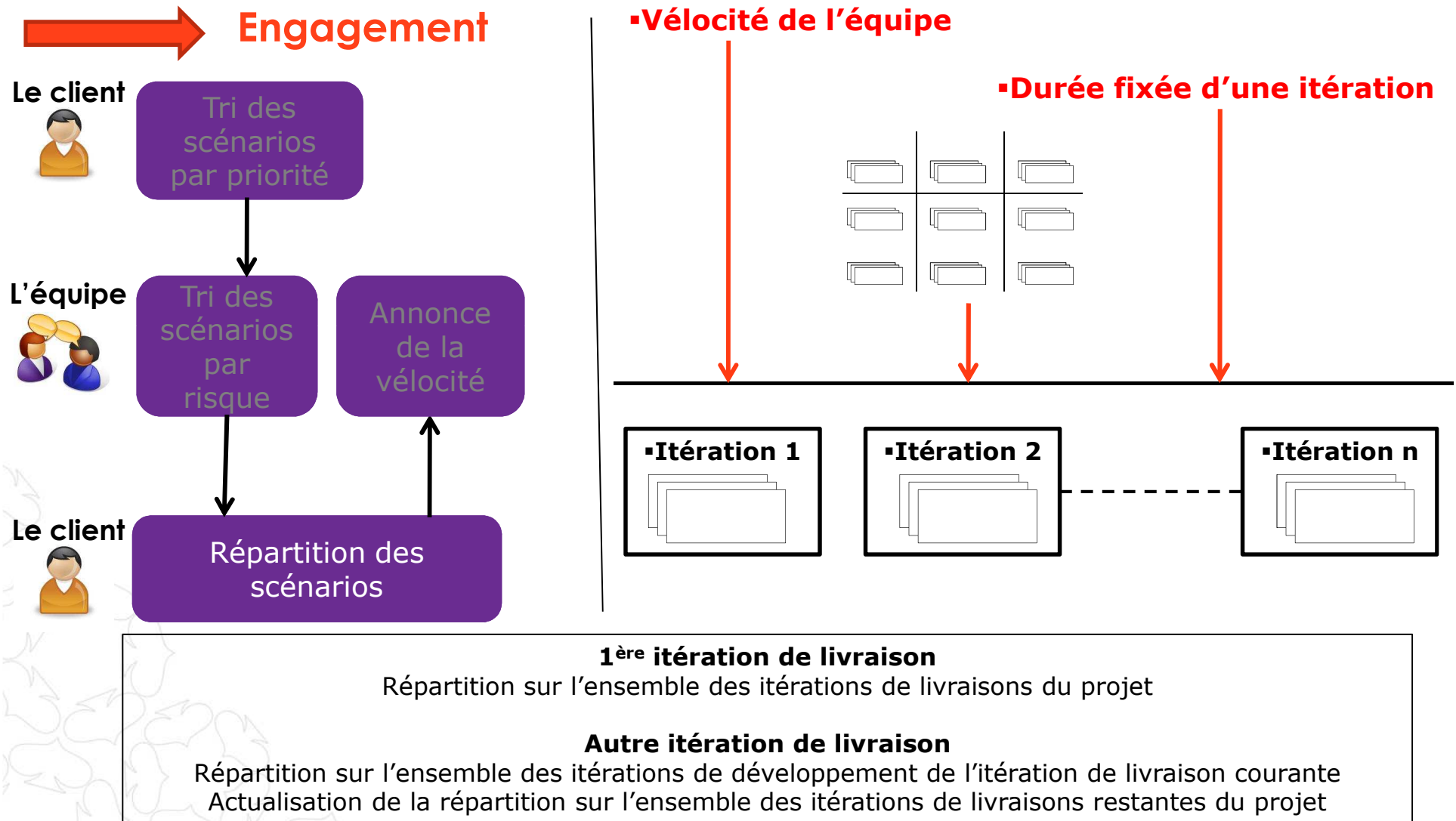
Exemple :

1 itération de 2 semaines
 1 point : 1 semaine de tps idéal
 1 binôme : 2 points par semaine
 1 développeur : 50% de programmation
 1 binôme : 2 pts dans 1 itération
 Equipe de 6 développeurs : 3 binômes : vélocité = 6

« ... En délivrant des fonctionnalités à chaque fin d'itération vous allez très rapidement pouvoir évaluer objectivement votre niveau de productivité et affiner itération après itération la charge et la date de fin prévue. »
 Freddy Mallet

La gestion de projet – Phases d'engagement

Répartition des scénarios



La gestion de projet – Les (sous) phases d'un développement

Exploration

Définition des tâches
(conception détaillée)

Le client



L'équipe



L'équipe



Engagement

Choix d'une tâche

Estimation
d'une tâche

Equilibrage

Evaluation de la vitesse
individuelle

Pilotage

La gestion de projet – La phase d'exploration

Exploration

Définition des tâches
(conception détaillée)

Le client



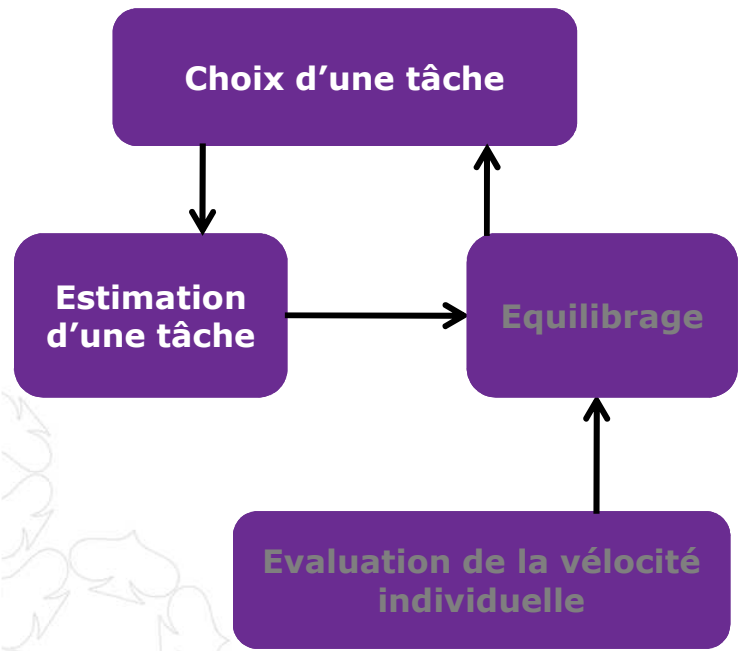
L'équipe



- **Tâche :**
 - Un groupement logique de travaux qu'un binôme peut développer en une ou deux journées
- **Décomposition d'un scénario en tâches**
- **Une tâche peut correspondre à une classe, un composant, ...**

La gestion de projet – La phase d'engagement - 1

L'équipe **Engagement**

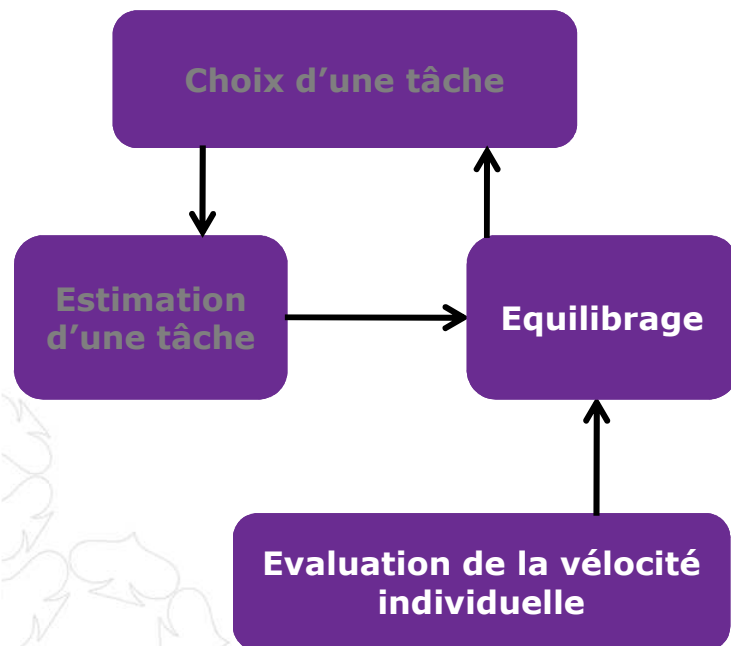


- Les programmeurs s'affectent la responsabilité des tâches
- Chaque programmeur estime la charges des tâches dont il est responsable en « jour-idéal »
- 1 « jour-idéal » = Une journée de travail hors « distraction » (café du matin, repas, pause, cigarette ? ...)

Tâche	Programmeur	Durée
Dessin de l'écran de saisie	Bf	1
Gestion de la saisie et de la mise à jour de la base	Jh	2
...		

La gestion de projet – La phase d'engagement - 2

L'équipe **Engagement**



- **Les programmeurs calculent leur charge totale**
 - Les tâches dont ils assurent la responsabilité
 - Les autres tâches en binôme
- **Redistribution des tâches pour atteindre un équilibre de charge entre les développeurs**
- **Scission / Fusion de tâches**
- **Ajout/ Suppression de scénarios client**
 - **Vélocité individuelle**
 - Nombre de points de scénarios client que le développeur peut traiter en une itération

La gestion de projet – La phase de pilotage

- **Le tracker**
 - Fait le point tous les 2 jours (+ ou -) avec chaque binôme
 - Relève par tâche le temps passé et le reste à faire
- **Le coach**
 - Réagit si nécessaire
 - Le triangle « infernal » contenu-ressource-délai
- **Les stand-up meetings**
 - Tous les matin un point rapide (10 mn)
 - Chaque programmeur s'exprime rapidement
 - Peut déclencher des réunions spécialisées
 - Ce n'est pas une réunion de suivi
 - C'est une réunion d'expression et d'échange
- **Clôturer le cycle**

- **Les pratiques de gestion de projet**

- Livraison fréquentes
- Planification itérative
- Client sur site
- Rythme durable

- **Les pratiques de programmation**

- Conception simple
- Remaniement
- Développement piloté par des tests

- **Les pratiques de collaboration**

- Programmation en binôme
- Responsabilité collective du code
- Règles de codage
- Métaphore
- Intégration continue



Cf. Chapitre
Ingénierie du
logiciel

SOMMAIRE

SCRUM

- Présentation
- L'équipe
- Backlog
- Planification
- Le fini
- Tests d'acceptation
- Spécialisation de Scrum
- Outils





- **Années 90** : Jeff Sutherland et Ken Schwaber définissent les grands principes
- **1996** : La méthode est publiée dans un premier article naissance
- **Définition** :
 - Une méthode agile pour améliorer la productivité des équipes
 - Une équipe soudée qui cherche à atteindre son but (mélée) en avançant individuellement (sprint)

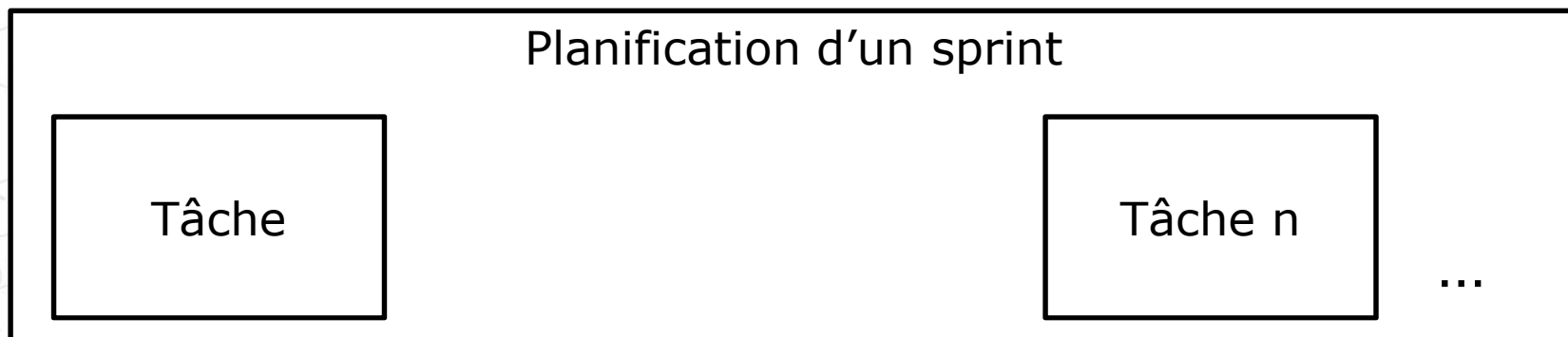
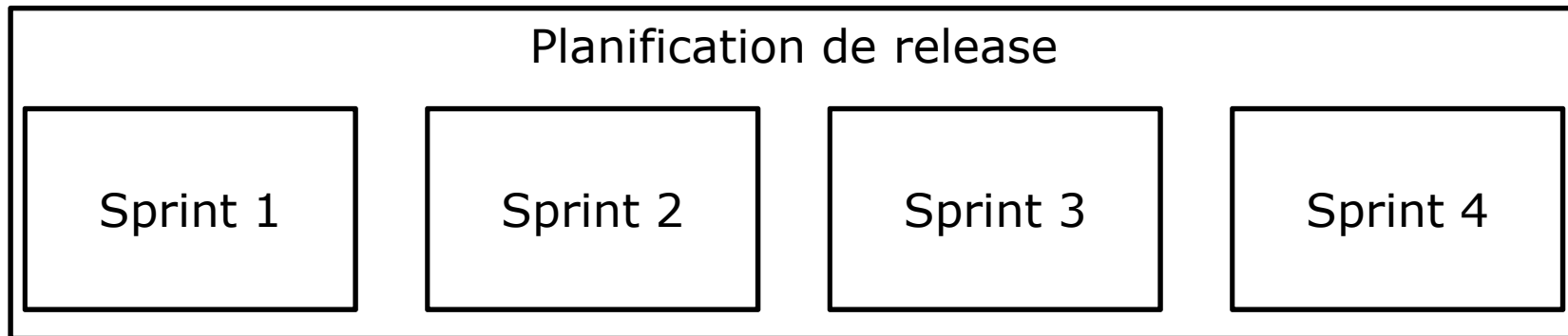
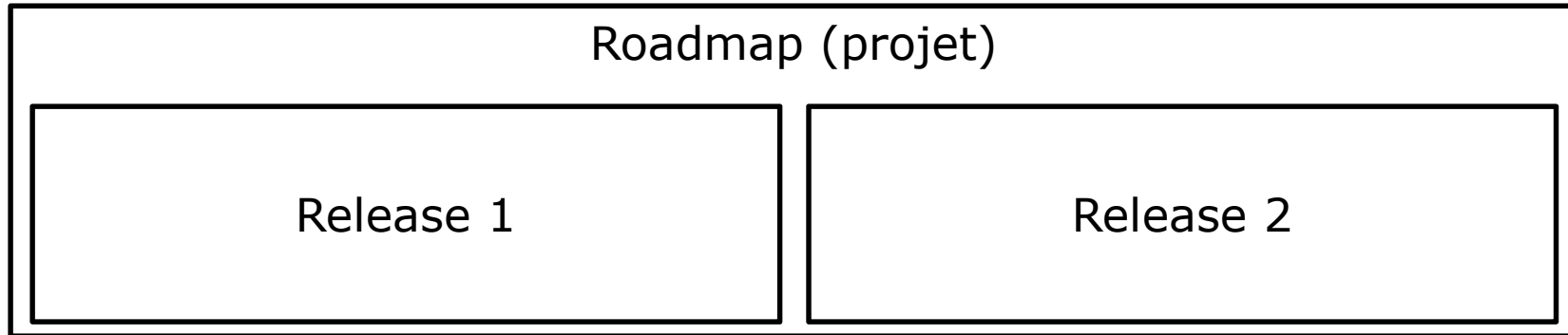
Présentation – Les valeurs de Scrum

- **Le client au cœur du projet**
- **Esprit d'équipe**
- **Importance de la communication**
- **Simplicité, efficacité et qualité**
- **Flexibilité au changement**
- **Avancement basé sur le concret**



We stand up to keep the meeting short.

Présentation – Démarche itérative



Présentation – Démarche



Vue synthétique du processus Scrum

Présentation – Quelques chiffres

- **Equipes pratiquant Scrum en France**

- **2005 : moins de 10 équipes**

- **2006 : moins de 50 équipes**

- **2007 : moins de 200 équipes**

- **2008 : moins de 800 équipes**

- **2009 : plus de 1 000 équipes**



We stand up to keep the meeting short.

■ Le Product Owner (Directeur de produit)

- Représente le client et les utilisateurs
- Définit les besoins
- Fixe les priorités
- Valide les solutions



■ L'équipe

- Multi compétences
- Auto gestion



■ Le Scrum Master

- Garant de la méthode
- Protège l'équipe des interférences extérieures

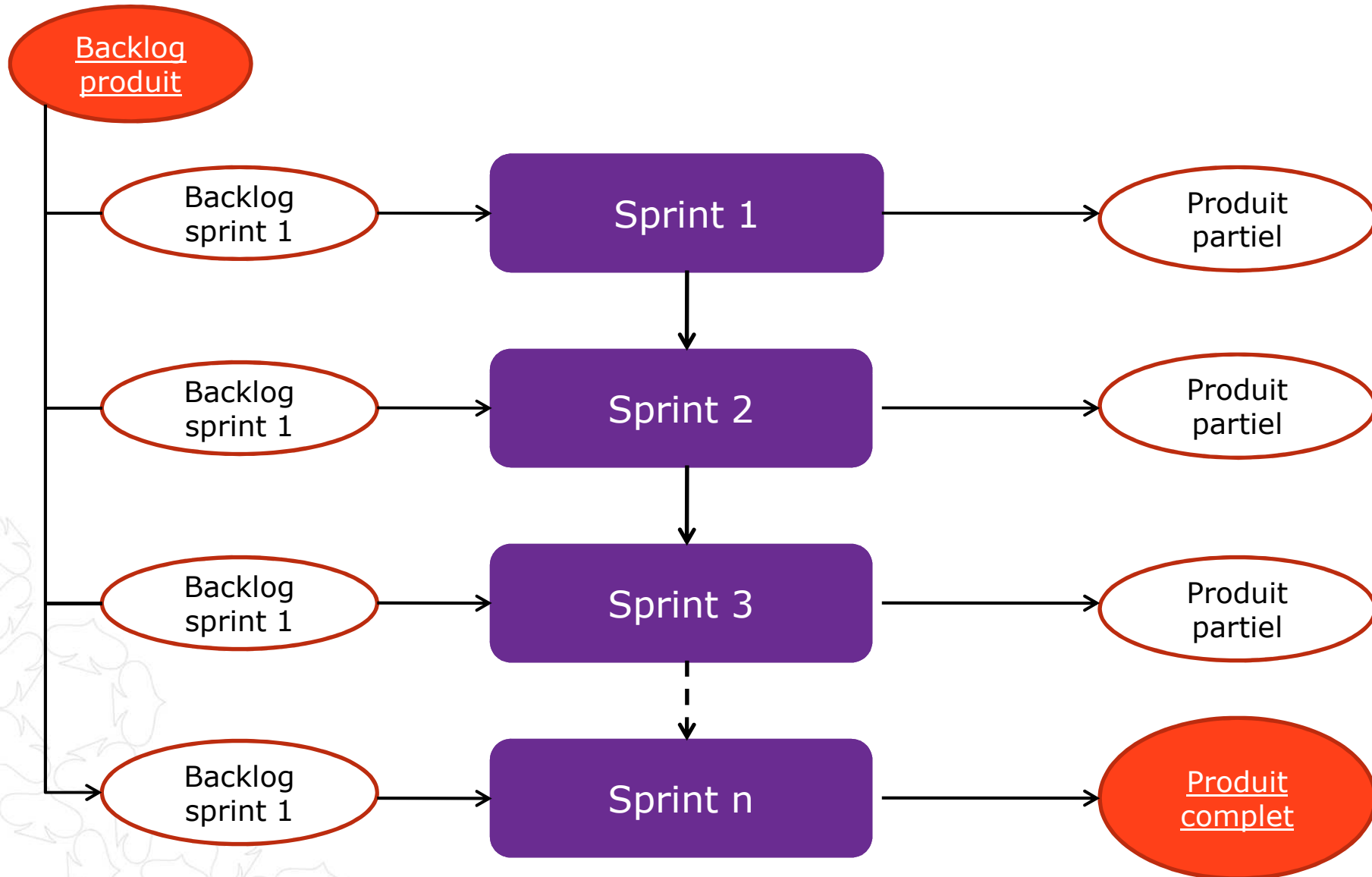


■ Les intervenants (Stakeholders)

- Les autres parties prenantes (Juridique, experts, architecte réseau...)



SCRUM Backlog



Backlog du produit

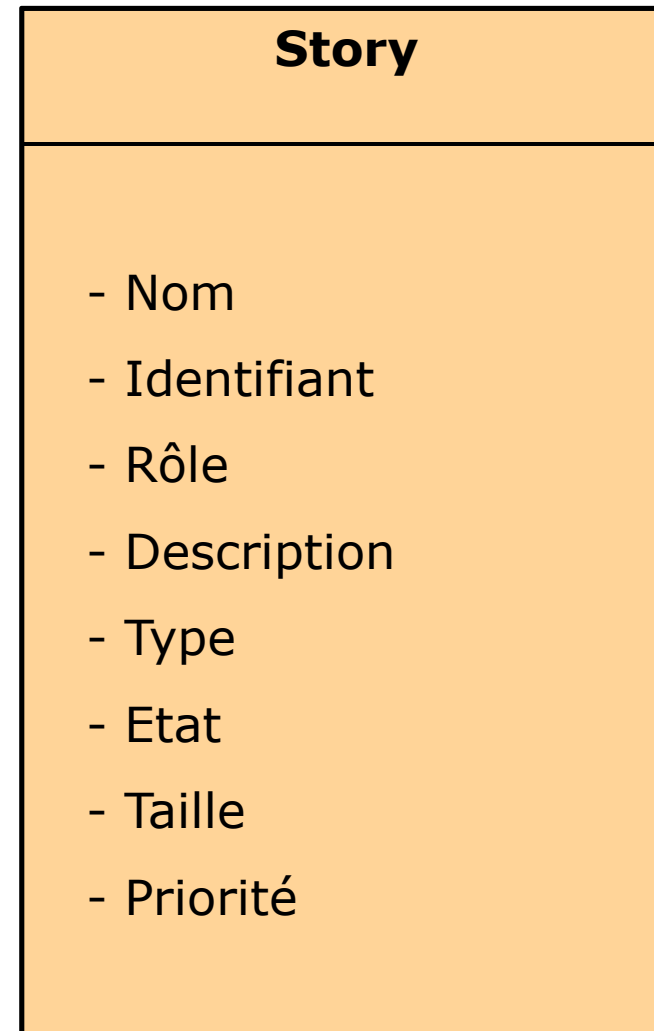
- **Une liste de fonctionnalités à réaliser**
 - Littéralement : la liste des choses en attente
 - Plus précisément : le référentiel des exigences
- **Chaque fonctionnalité est décrite par une story**
 - Les stories sont rangées par priorité
- **Le backlog produit est défini en amont du projet Scrum**
 - Les fonctions essentielles sont identifiées
 - Features, macro stories
- **Le backlog est actualisé tout au long du projet**
 - La description des fonctions est affinée au fil du projet
 - Features, macro stories, stories

Backlog – Story :

Élément du backlog du produit

Story :

- Un élément fonctionnel qui apporte une valeur et qui peut être développé en un sprint.
- Story est un concept de XP.



Backlog – Story :

Élément du backlog du produit

Story
<ul style="list-style-type: none">- Nom- Identifiant- <u>Rôle</u>- Description- Type- Etat- Taille- Priorité

Permet de préciser **le rôle de l'utilisateur**.

Cet attribut peut aider à la **compréhension** de la story.

Il permet aussi de faciliter la recherche d'exhaustivité de stories.

Exemple : Le client, le responsable des stocks, la comptabilité,...

Chaque rôle peut donner lieu à une **description** spécifique en termes :

- Description du rôle
- Fréquence d'utilisation du produit
- Nombre de personnes ayant ce rôle
- Ses critères de satisfaction dans l'utilisation du produit

Cf. Use case d'UML

Backlog – Story :

Élément du backlog du produit

Story

- Nom
- Identifiant
- Rôle
- **Description**
- Type
- Etat
- Taille
- Priorité

Description libre

Quelques lignes

Ou **structurées**

Exemple

Rôle de l'utilisateur
(en tant que)

But
(Je veux)

Justification
(afin de)

En tant que client, **je veux** m'inscrire à une session de formation **afin** d'obtenir une nouvelle compétence

En tant que client je **veux** acheter un billet de train **afin** de me rendre d'une ville à une autre

Backlog – Story :

Élément du backlog du produit

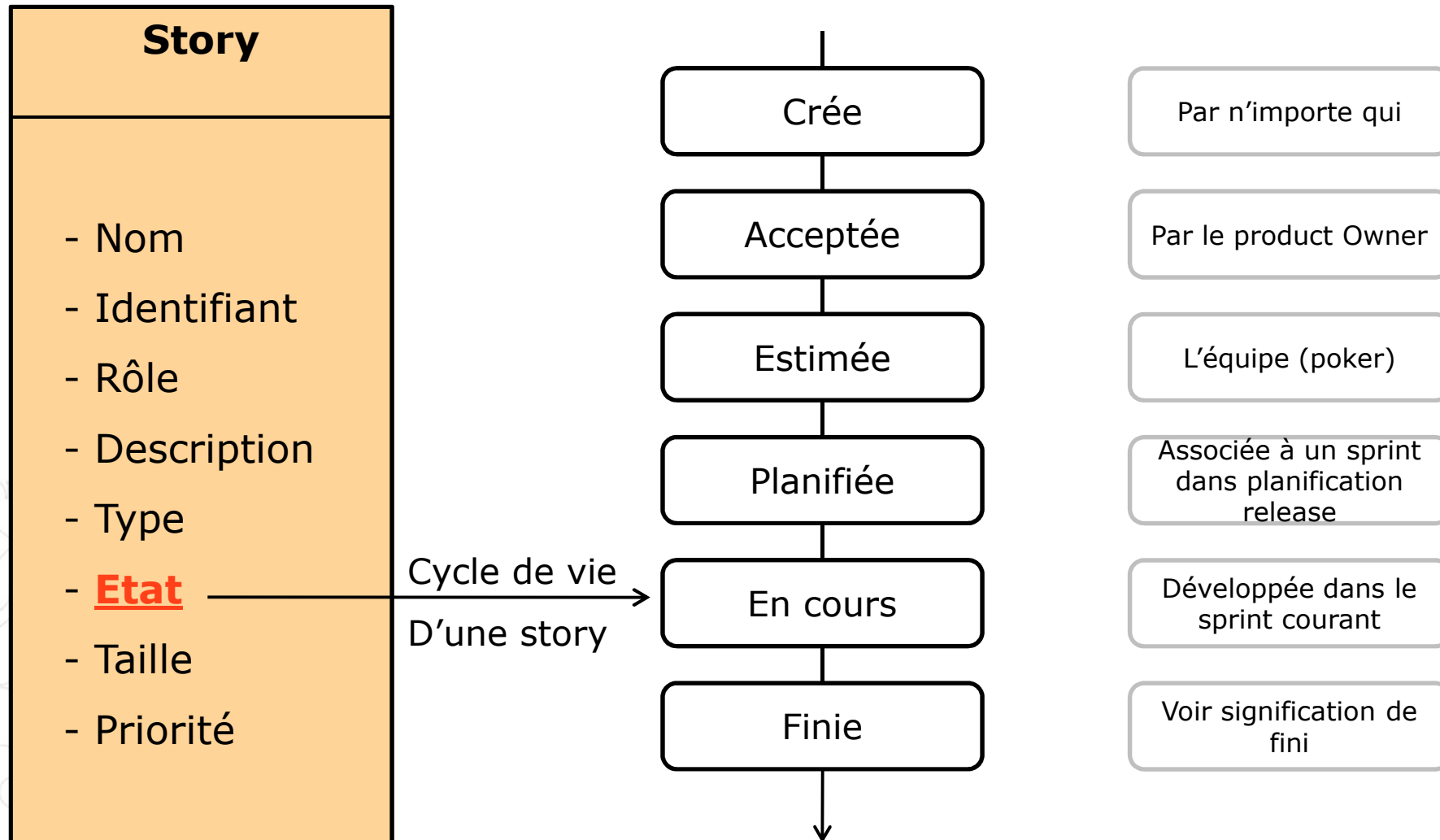


User story : vue utilisateur

Story technique : vue développeur

Défaut : bug

Backlog – Story : Élément du backlog du produit



Backlog – Story :

Élément du backlog du produit

Story
- Nom
- Identifiant
- Rôle
- Description
- Type
- Etat
- <u>Taille</u>
- Priorité

Petites tailles :

- Les stories prioritaires
- Prêtes à développer
- Seront associées à des séries de tests

Tailles moyennes et grandes :

- Pourront être décomposées



Priorité 1, 2, 3,

Evaluer la taille : Le planning Poker

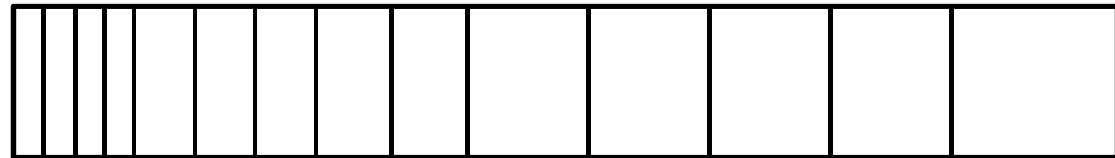
Backlog – Story :

Élément du backlog du produit

Story

- Nom
- Identifiant
- Rôle
- Description
- Type
- Etat
- Taille
- **Priorité**

Les stories d'un backlog sont classées par priorité



Priorité 1, 2, 3,

Les critères pour fixer les priorités

- La réduction des risques
- La diminution de l'incertitude sur des besoins
- L'augmentation de la qualité
- Les dépendances entre stories

Evaluer la priorité : le priority Poker

SCRUM

Backlog – Story : Identification

- **Une approche par les données**
 - **Gestion client**
 - **Gestion fournisseur**
 - **Gestion commande**
 - ...

- **Une approche par les processus**
 - **Traitement d'une commande**
 - **Traitement d'une livraison**
 - ...



SOGETI France

22-24, rue du Gouverneur Général Eboué
92136 Issy les Moulineaux
www.sogeti.com