

Le développement Agile

Yannick Mazières

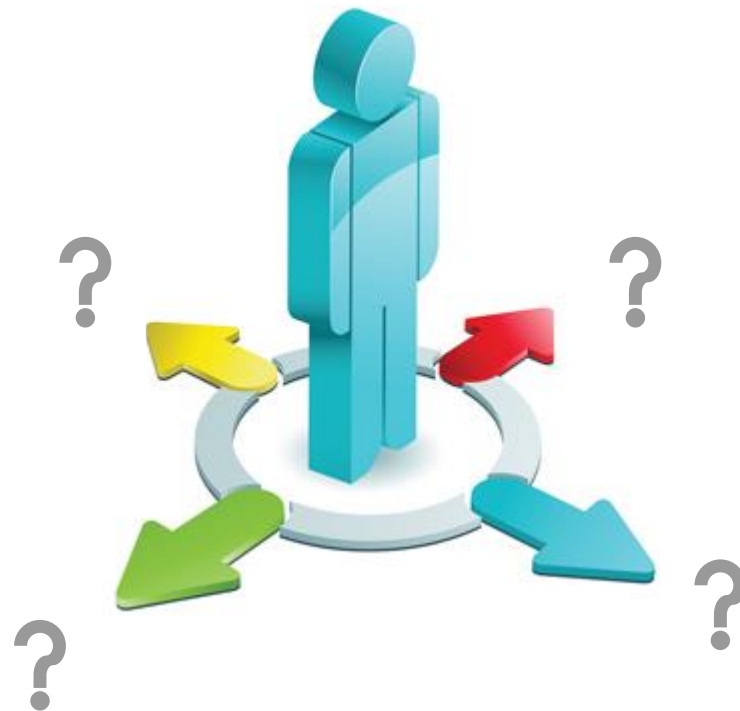
Département informatique

Cégep Sainte-Foy

Trois vérités sur la gestion de projet

1

C'est impossible de connaître tous les besoins au début d'un projet.



Trois vérités sur la gestion de projet

2

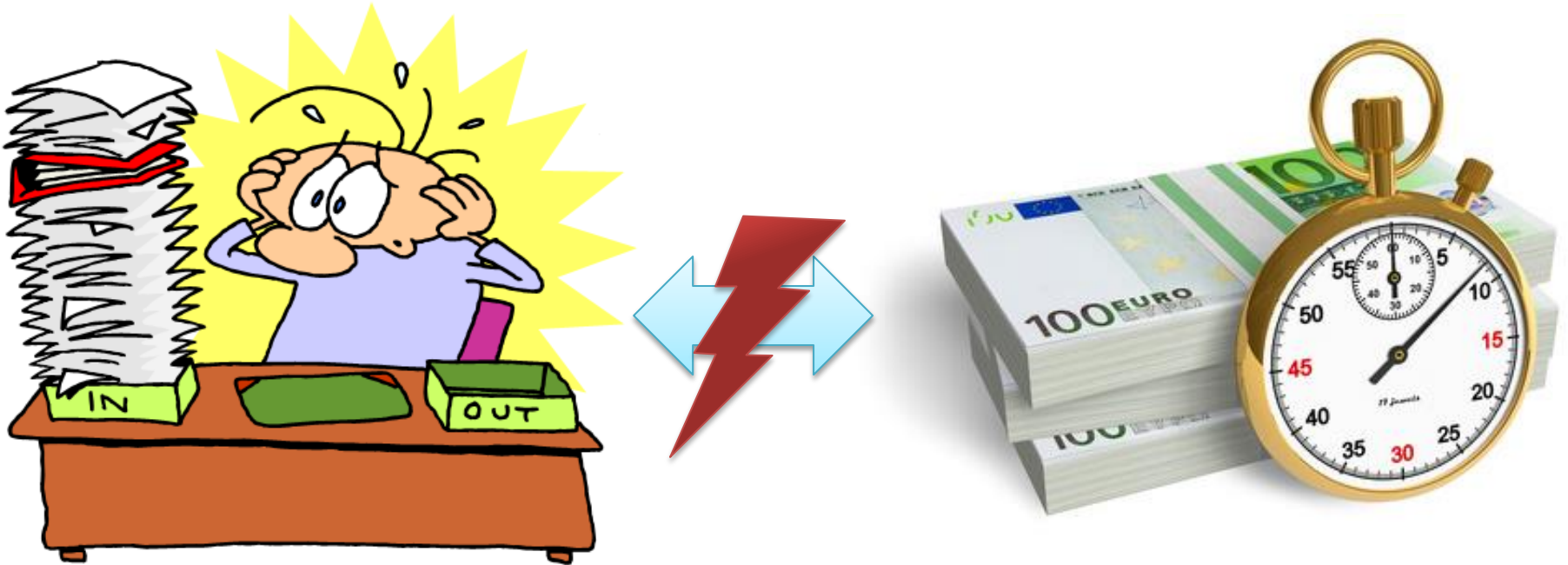
Peu importe les besoins identifiés,
il est assuré qu'ils changeront dans le temps.



Trois vérités sur la gestion de projet

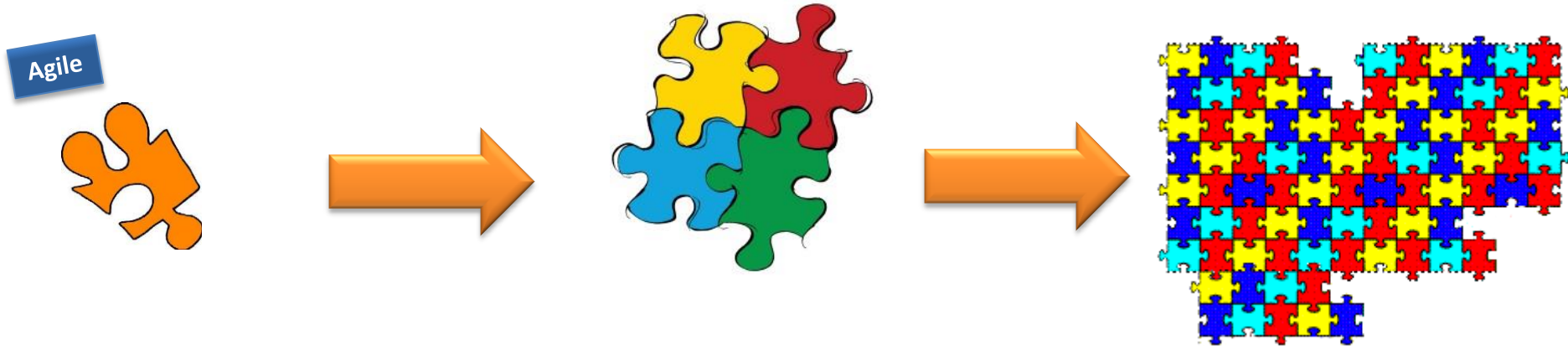
3

Il y'aura toujours plus à faire qu'il y'a de temps et d'argent disponible.



Agile VS Traditionnelle

L'agilité est une approche de développement logiciel itératif, incrémental et adaptatif...



...plus que le suivi d'un plan



Agile VS Traditionnelle

Traditionnelle

Analyse

Design

Code

Test

Plus difficile de s'adapter aux changements

Les fonctionnalités sont développées plus tard dans le processus. Les besoins identifiés ont-ils changés depuis le début ?

Qu'est-ce qu'on coupe si on manque de temps ?

Malheureusement, c'est l'assurance de la qualité qui risque d'être coupée

Agile

Itération 1

Itération 2

Itération 3

Analyse

Analyse

Analyse

Design

Design

Design

Code

Code

Code

Test

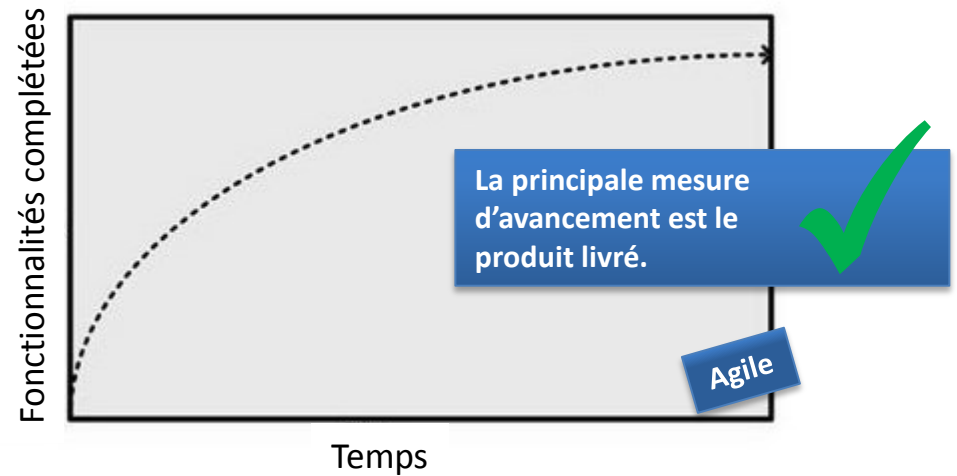
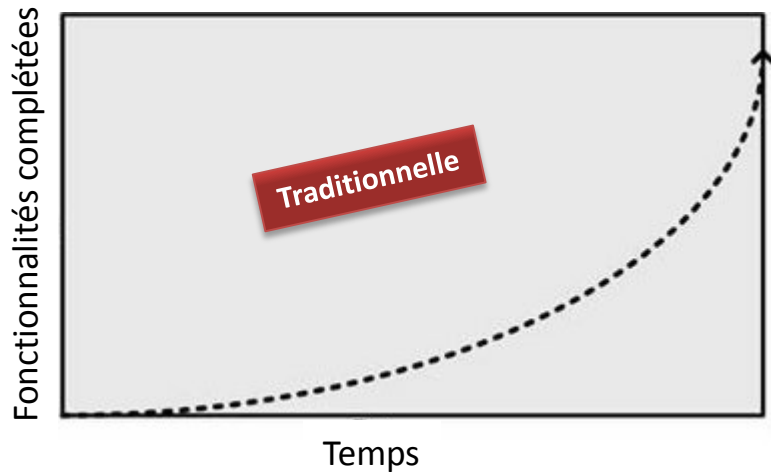
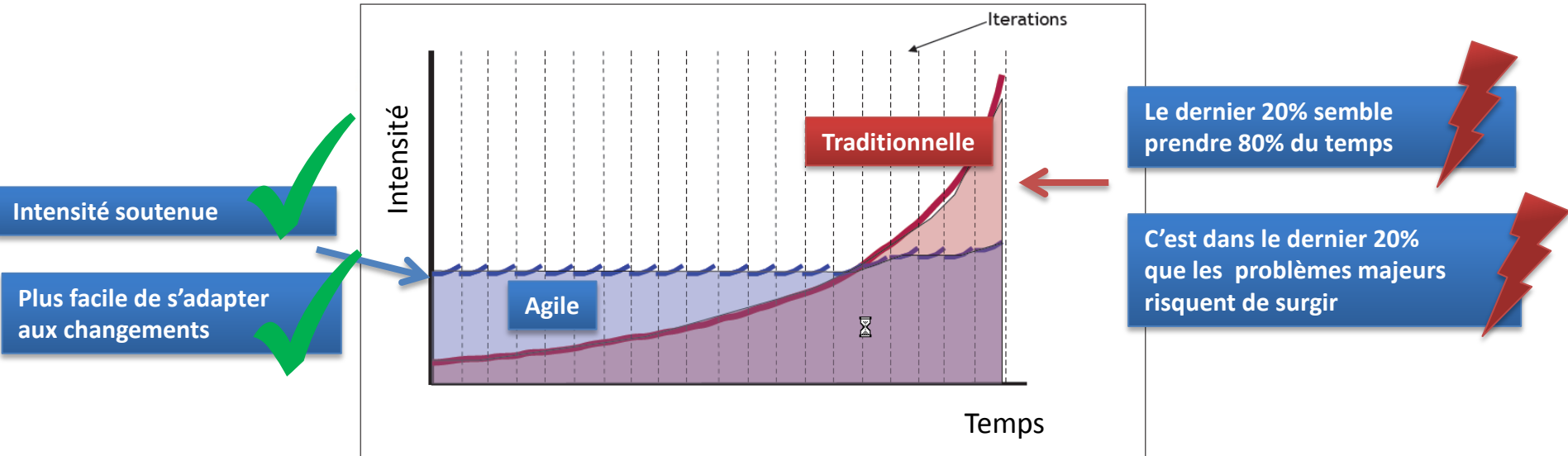
Test

Test

Plus facile de s'adapter aux changements

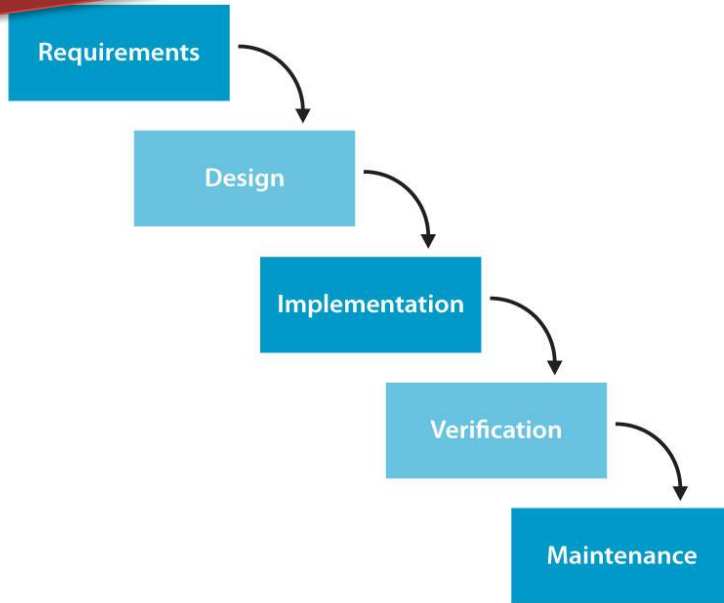
La qualité est assurée tout le long du processus

Agile VS Traditionnelle



Agile VS Traditionnelle

Traditionnelle



Les approches **Waterfall** sont **coûteuses** et sont très **risquées** dans des environnements très **changeants**.

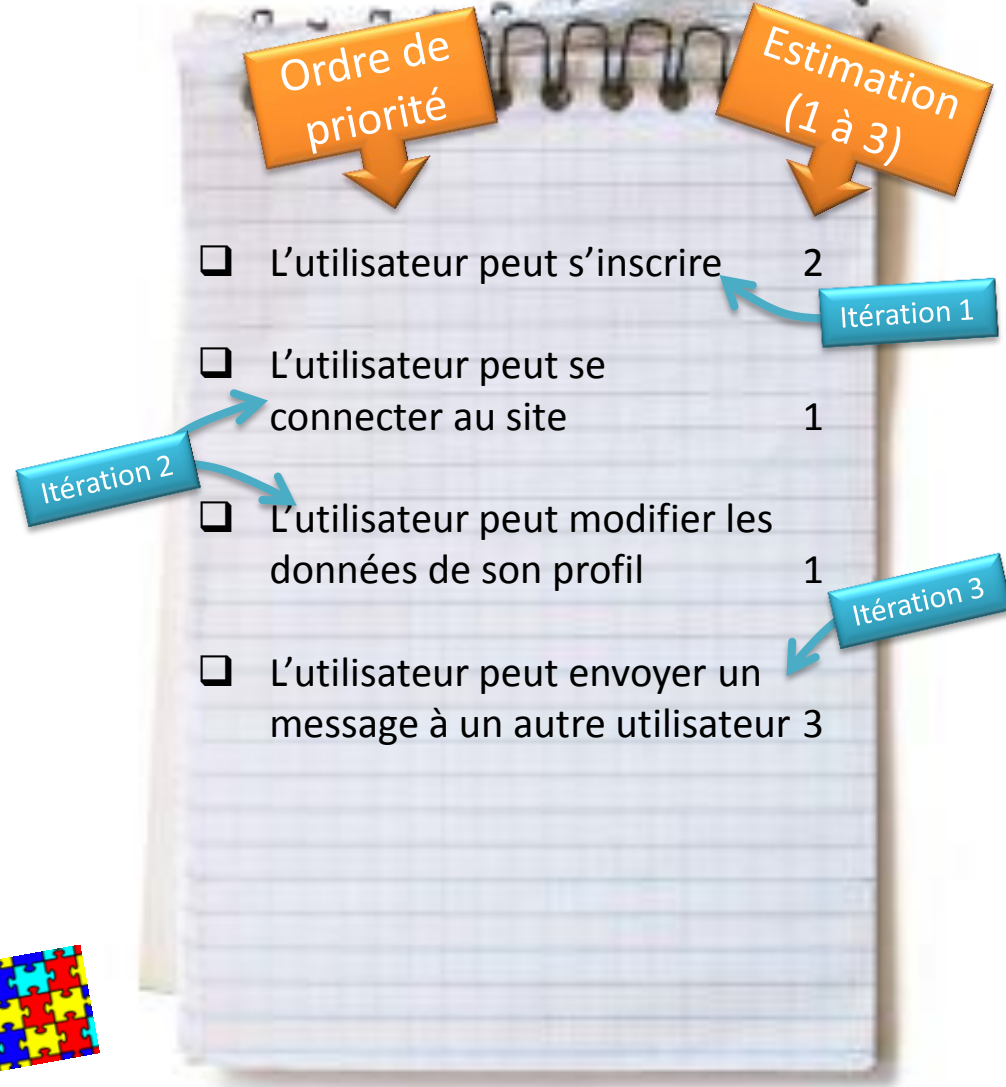
Approche efficace quand :

- Les requis sont stables (contexte bien établi qui ne change pas)
- Il existe peu d'inconnus et de nouveautés
- La technologie est mature et bien connue
- Tout se déroule comme prévu
- Plusieurs projets semblables ont déjà été réalisés

Agile c'est quoi ?

Pour y arriver, il faut:

1. Identifier les **User Stories**
(fonctionnalités du point de vue de l'utilisateur)
2. **Estimer** la complexité de chacun des **User Stories**
3. Prioriser les **User Stories**
4. Développer et livrer des **User Stories (intégration continue)** aux deux semaines (**itérations**)



Les User stories



VS



Un **User Story** doit exprimer une fonctionnalité à développer du point de vue de l'utilisateur (et non du développeur).

Les **User Stories** complétés permettent de mesurer l'avancement du projet.

L'estimation et la vélocité

L'estimation se fait en comparant les tâches entre elles et en appliquant une valeur (des points) à la complexité d'un **User Stories**.

La **vélocité** est le nombre de points qui peut être réalisé par l'équipe en **une itération**.

Estimation
(1 à 3)

☒ L'utilisateur peut s'inscrire 2

☒ L'utilisateur peut se connecter au site 1

☒ L'utilisateur peut modifier les données de son profil 1

☒ L'utilisateur peut envoyer un message à un autre utilisateur 3

☐ L'utilisateur peut importer un dossier 3

Itération 1
2 semaines

Vélocité de 3

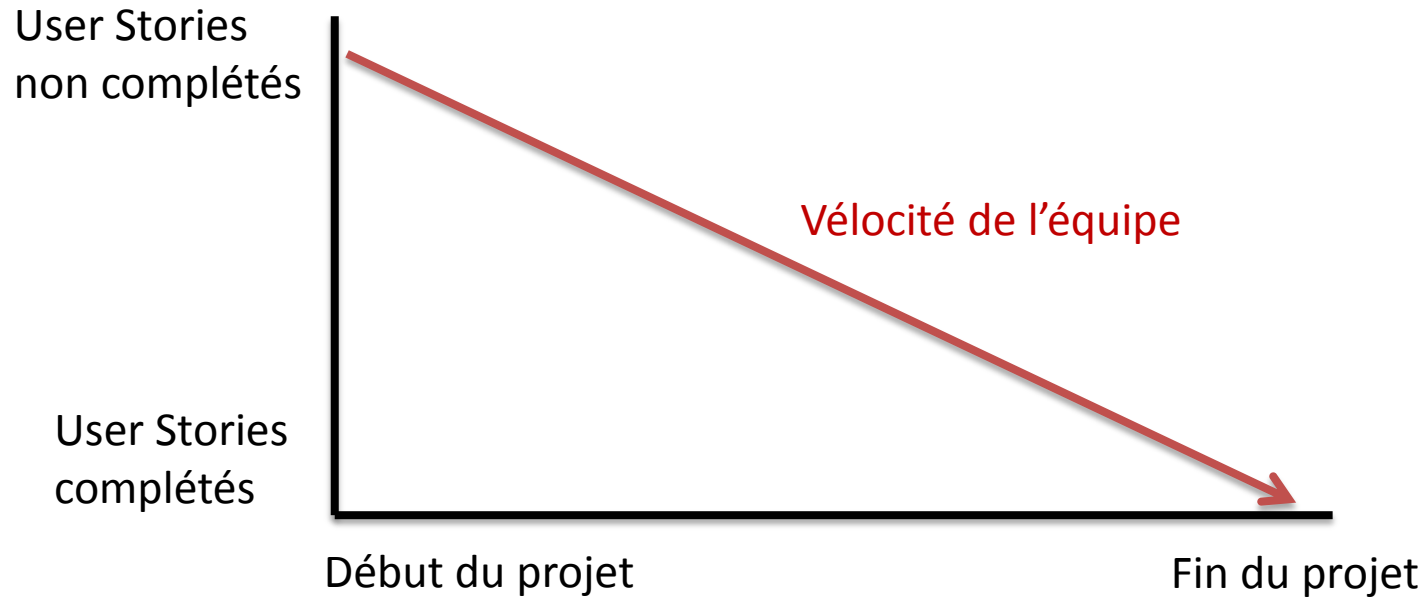
Itération 2
2 semaines

Vélocité de 4

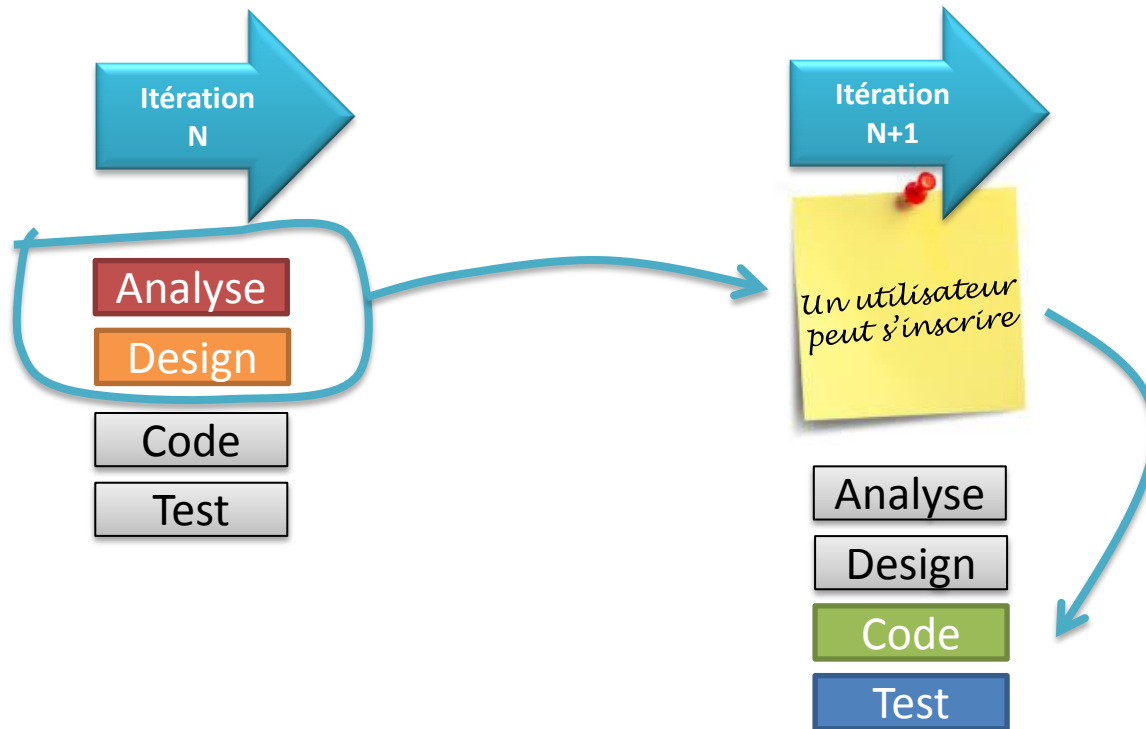
Itération 3
2 semaines

Devrait se réaliser en une itération car la vélocité moyenne est de 3.5

L'estimation et la vélocité



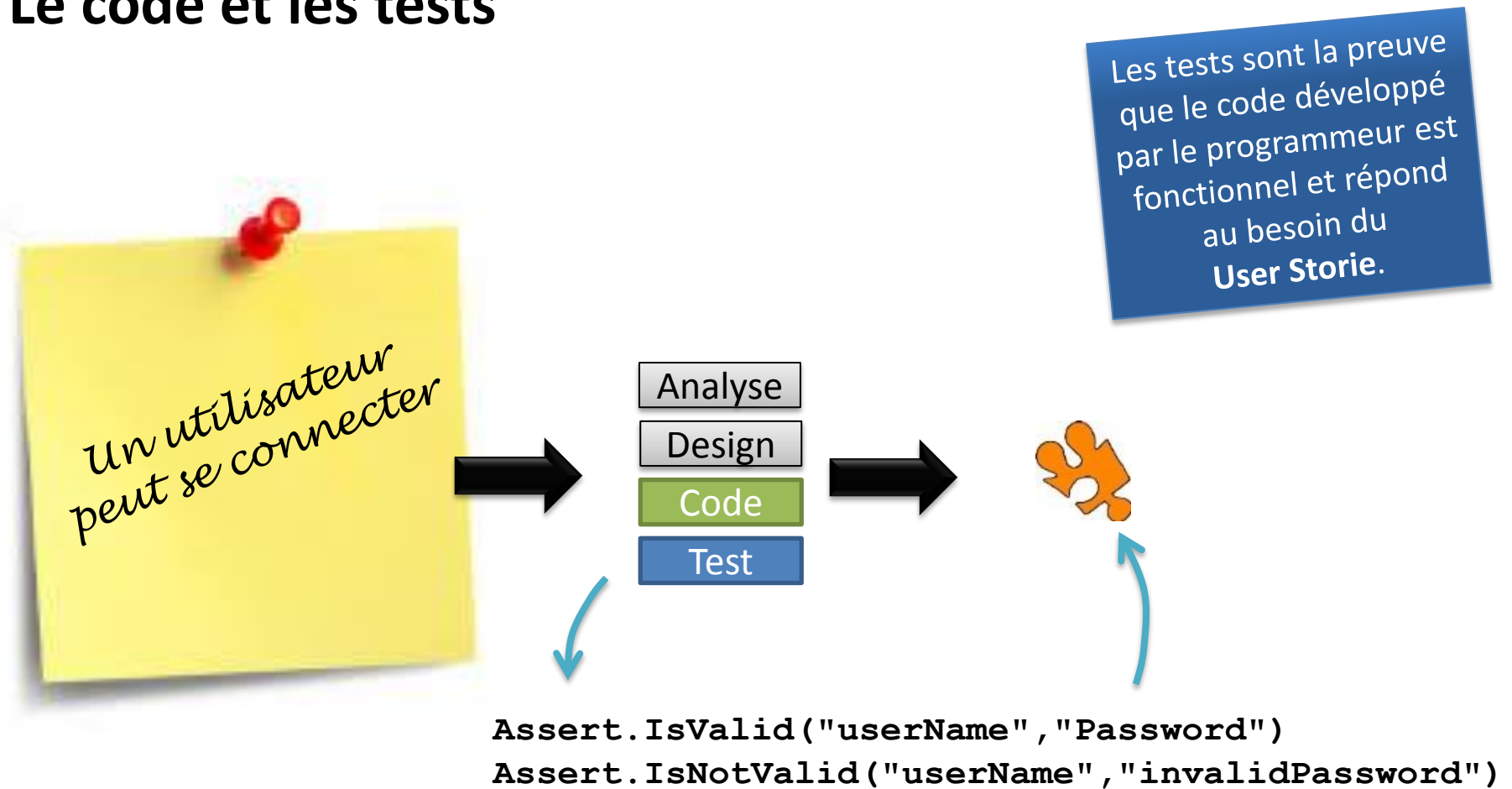
L'analyse et le design



Note:
Selon le contexte, l'analyse
peut très bien se faire dans la
même itération (au lieu de la
devancer comme dans
l'exemple ci-contre)
Il faut s'adapter !

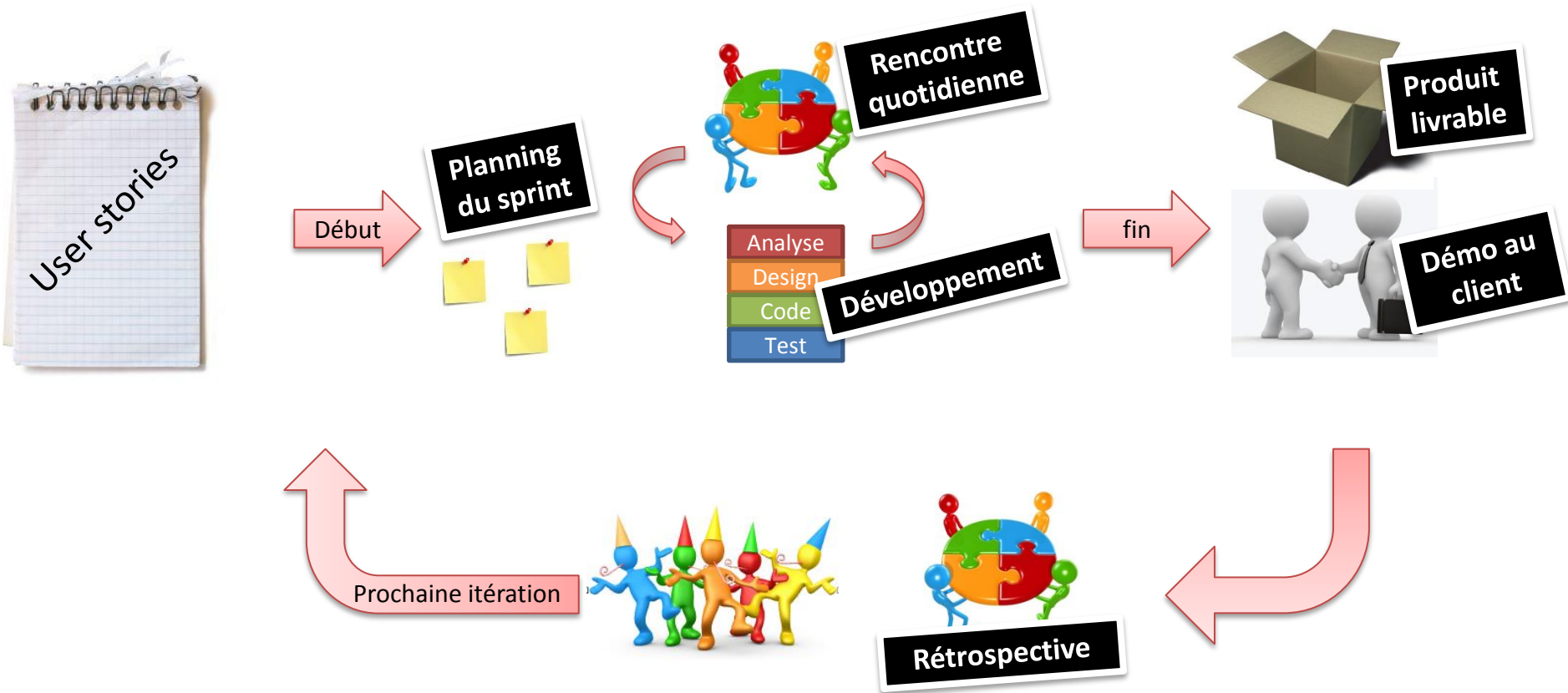
- L'analyse d'un **User Storie** est adapté selon le contexte.
Exemples: structure de l'entreprise, ampleur du projet, etc.
- Les techniques **d'analyse et de design** peuvent être nombreuses (diagramme de cas d'utilisation, diagramme de flux de données, description de scénarios, prototype sur papier, description des utilisateurs, maquettes, etc.).
- Les techniques et outils d'analyse doivent être **légers et souples** et non **lourds et bureaucratiques**. Ils doivent permettre de **s'adapter** aux changements.

Le code et les tests



Cycle d'une itération (un sprint)

Règle générale une itération à une durée de 2 semaines mais peut varier selon le contexte.



Les user stories proviennent de:

- L'équipe
- Management
- Direction
- Clients
- Usagers (forum, feedback, tests, etc.)

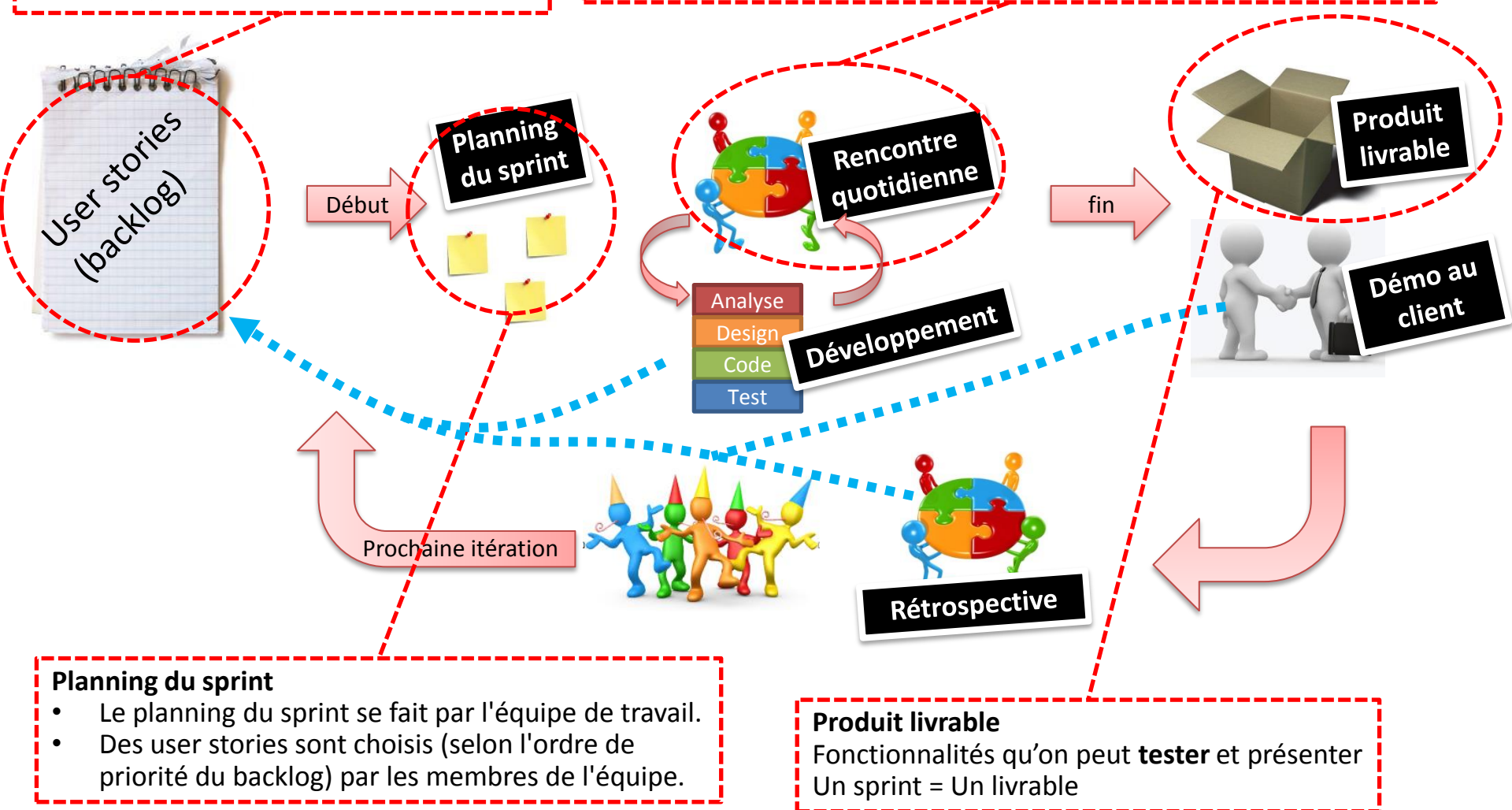
On peut ajouter / modifier des éléments du backlog à tout moment

Rencontre quotidienne

Pour chacun des membres de l'équipe, répondre aux trois questions suivantes:

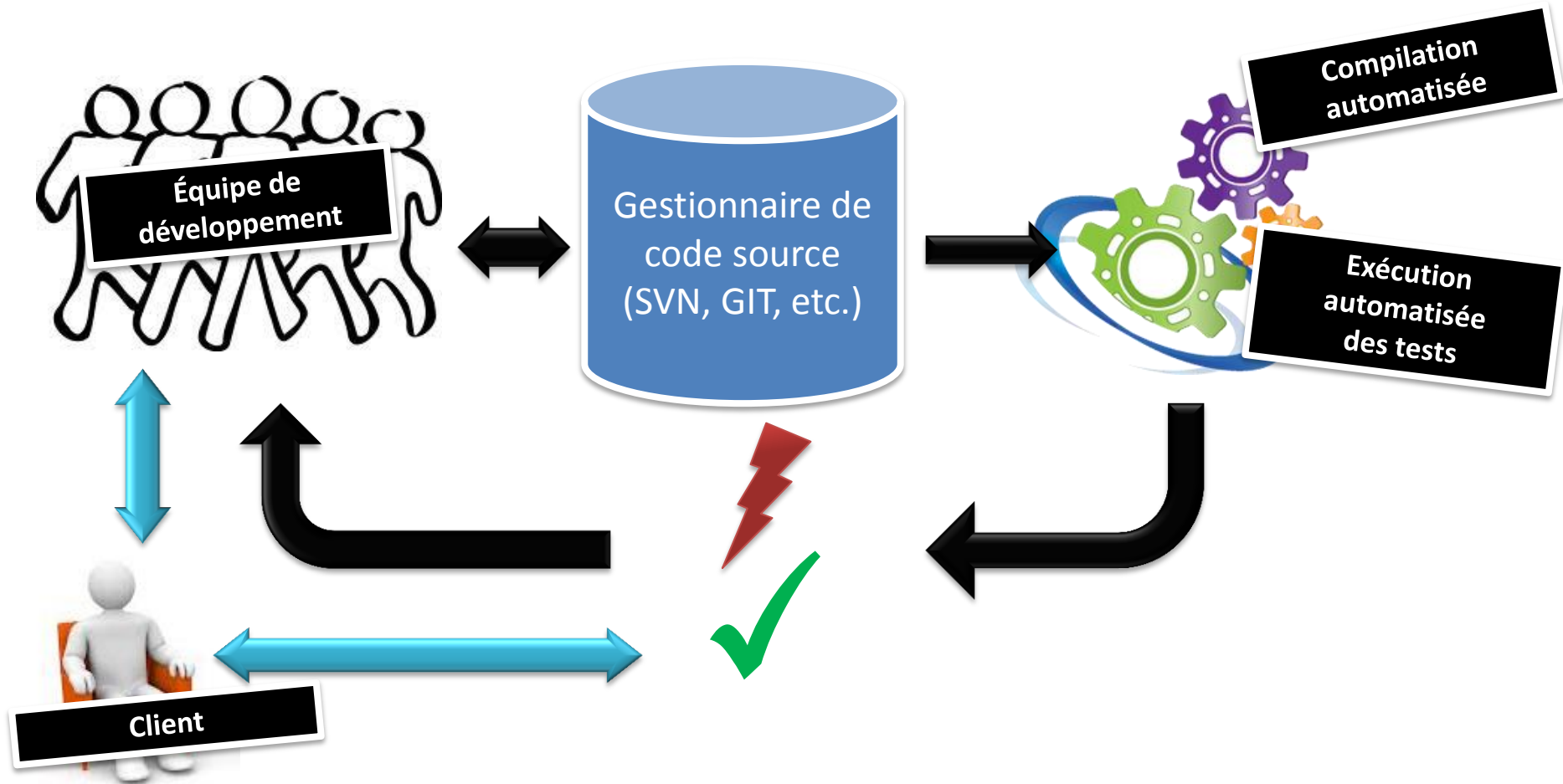
- *Qu'est-ce que j'ai fait hier ?*
- *Qu'est-ce que je compte faire aujourd'hui ?*
- *Quelles sont les difficultés que je rencontre ?*

Chacun des membres de l'équipe est responsable de mettre à jour ses user stories.



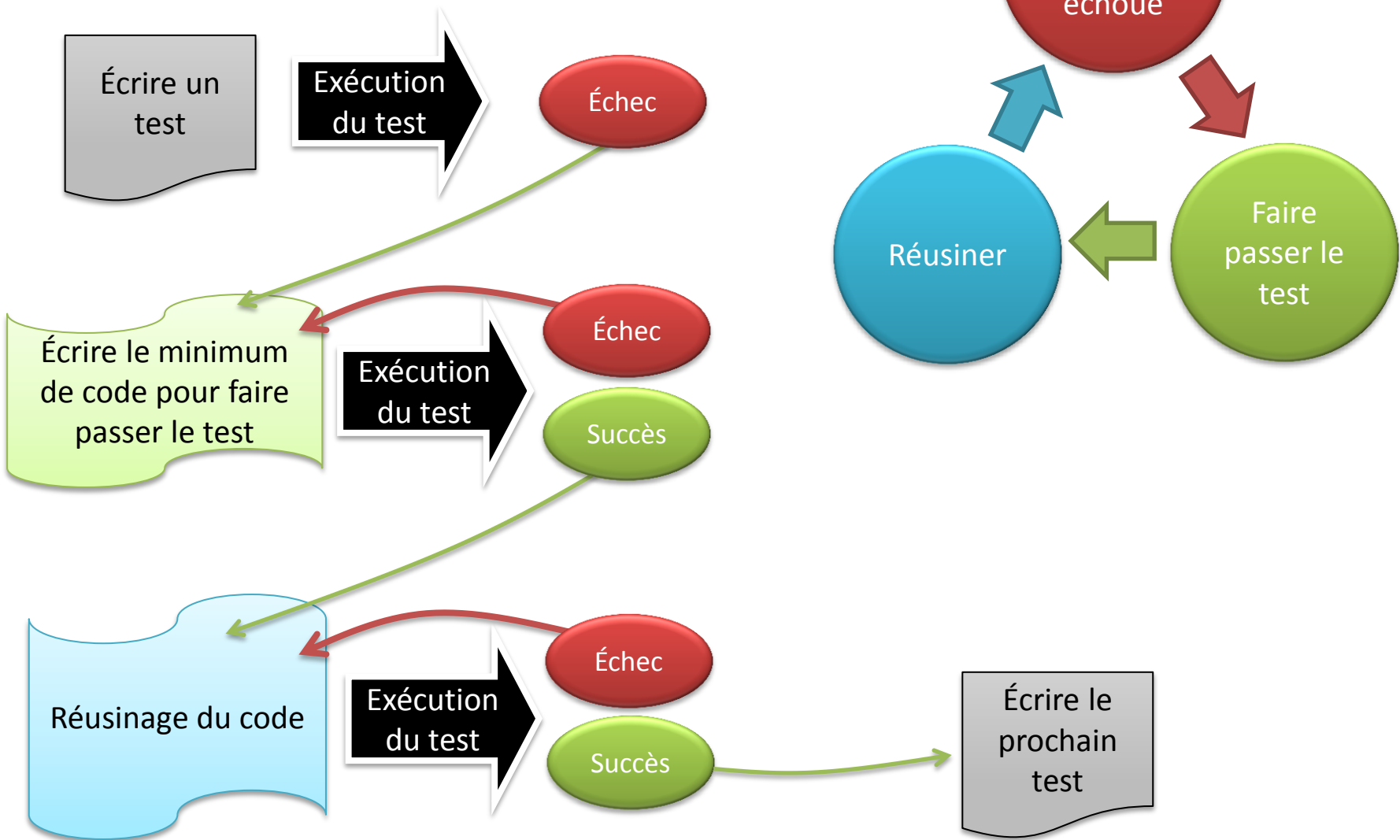
Des outils et techniques agiles

Intégration continue



Des outils et techniques agiles

TDD (test driven development)



Des outils et techniques agiles

Outils de planification

Le projet

**Tous les User Stories
(backlog)**



Une itération

**User Stories
à faire**



**User Stories
en développement**

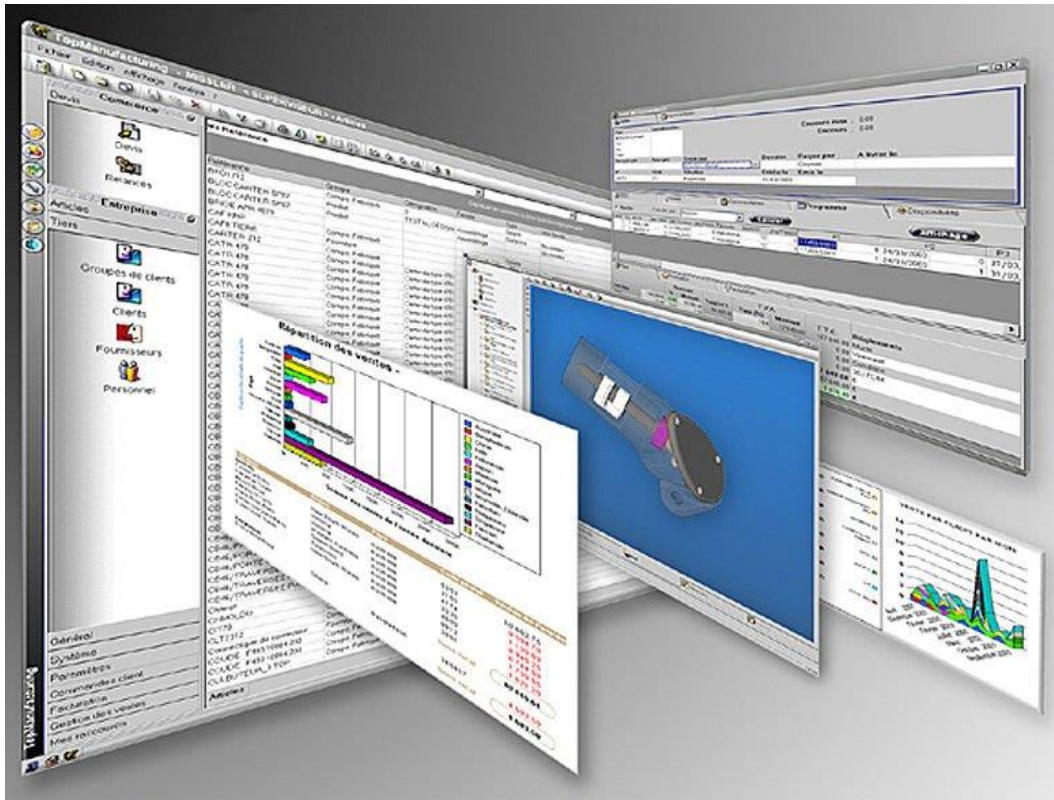


**User Stories
Complétés**



4 valeurs (manifeste agile¹)

Des logiciels opérationnels
plus qu'une documentation exhaustive.



¹ Texte rédigé en 2001 par 17 experts du développement d'applications informatiques

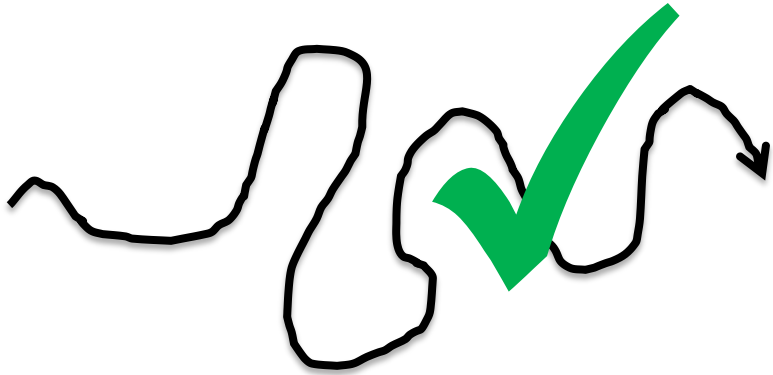
4 valeurs (manifeste agile)

La collaboration avec les clients
plus que la négociation contractuelle.



4 valeurs (manifeste agile)

L'adaptation au changement
plus que le suivi d'un plan.



4 valeurs Agile(manifeste agile)

Les individus et leurs interactions
plus que les processus et les outils.



Les processus et
outils doivent être
légers et souples et
non lourds et
bureaucratiques

Quelques principes Agile

Satisfaire le client en **livrant rapidement des fonctionnalités** (exemple: aux 2 semaines).

Accueillir positivement les **changements de besoins**, même tard dans le projet.

Utilisateurs et développeurs doivent travailler ensemble tout au long du projet.

Un **logiciel opérationnel** est la **principale mesure d'avancement**.

La simplicité est essentielle (l'art de minimiser la quantité de travail inutile) . Exemple: automatisation des tâches.

À intervalles réguliers, l'équipe **réfléchit aux moyens de devenir plus efficace**, puis règle et modifie son comportement en conséquence.

Être Agile c'est:

- Livrer chaque semaine un produit de qualité avec une valeur ajoutée.
- Réfléchir sur comment être plus efficace et pouvoir s'améliorer (s'adapter aux changements).



Inspiré du livre:

The Agile Samurai

How Agile Masters Deliver Great Software
by Jonathan Rasmusson

