



La philosophie Agile

UN SET DE VALEURS POUR UNE MEILLEURE COLLABORATION

A propos de moi

- ▶ Kevin Glass, PhD
- ▶ 15 ans d'expérience en développement informatique
- ▶ CTO d'une entreprise SaaS

“

Pendant mon dernier projet on
faisait ...

”

RACONTEZ MOI VOS EXPÉRIENCES !

Avez-vous déjà travaillé en groupe ? Cela s'est passé comment ?

“

Je me souviens quand j'ai fait un
projet de jeux-vidéo pour une
musée...

”

Je vous raconte mon expérience en tant que Lead Développeur pour un parcours de jeux-vidéo dans un musée en France

La promesse de l'Agile

Projets livrés à temps

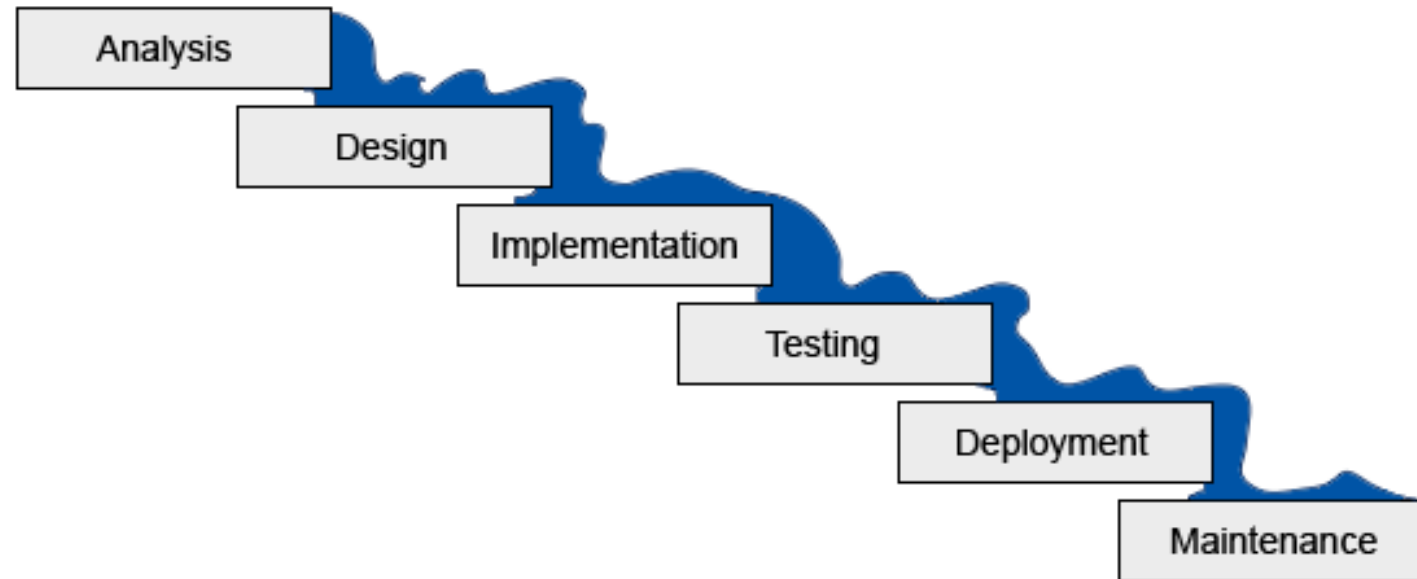
Meilleure qualité

***Performance et
productivité
époustouflante !***

Utilisateurs satisfaits

Equipe heureux et qui travail des horaires
« normaux »

La méthode notoire « Waterfall »



La méthode notoire « Waterfall »



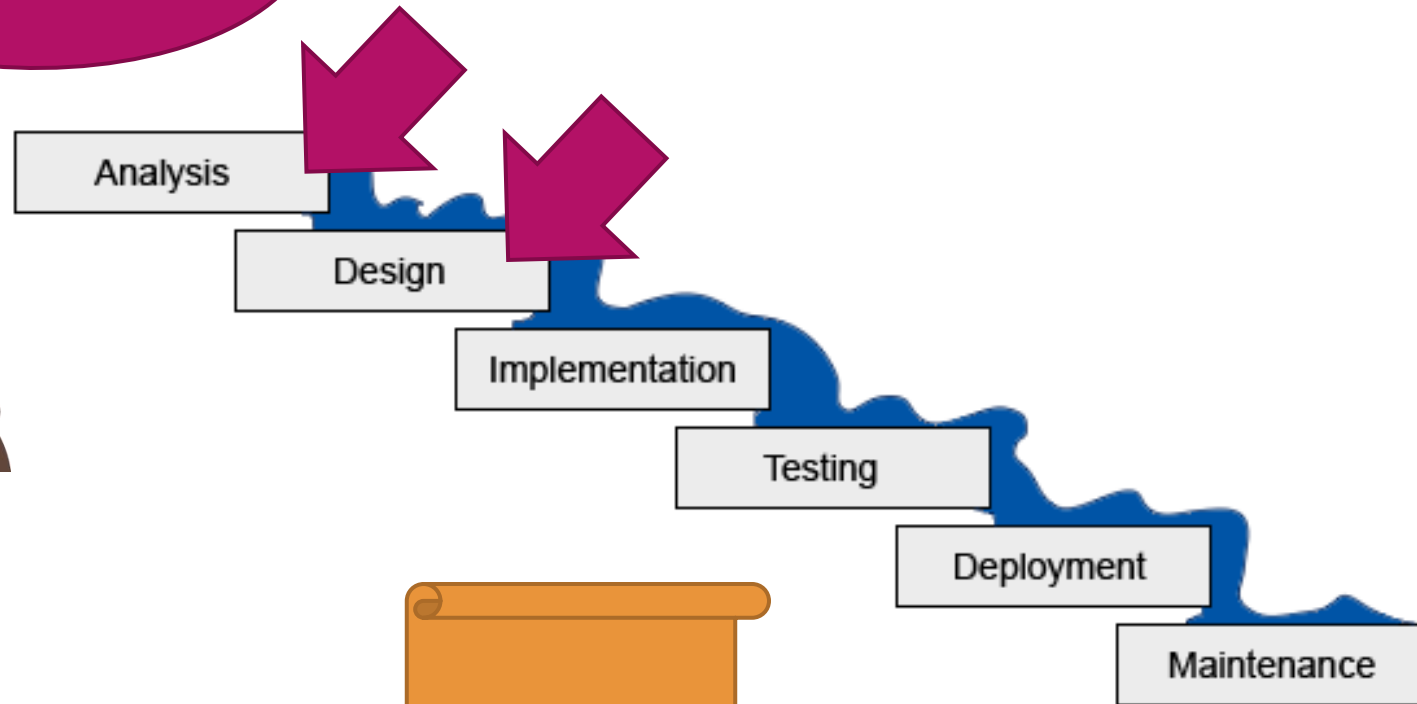
CEO

Le projet **VR** est en route, j'assure mes investisseurs

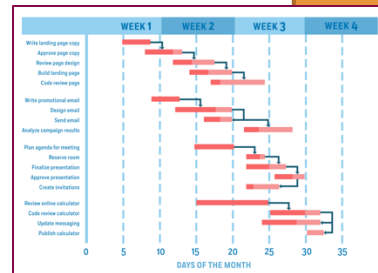


Project Manager

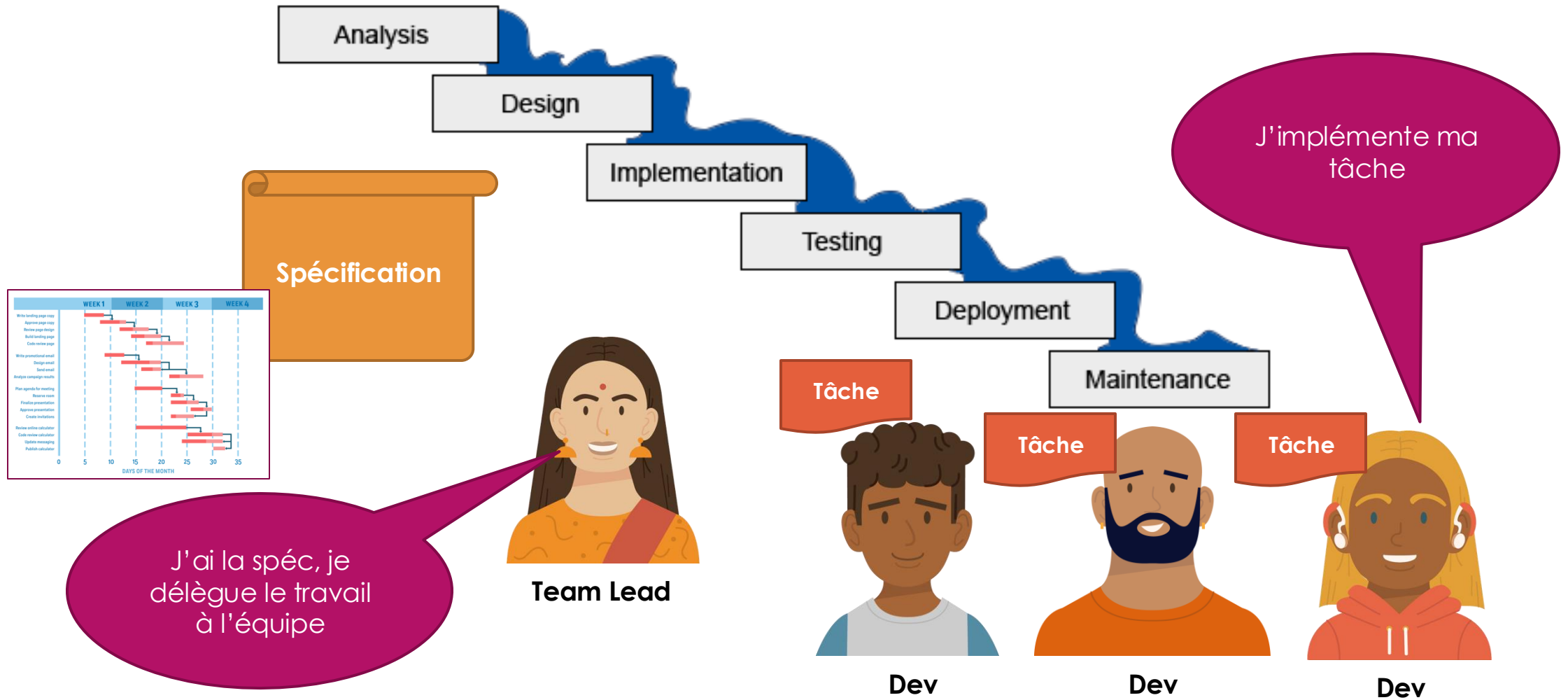
J'ai parlé aux clients, et j'ai créé la spécification et le plan qui dur 2 ans



Spécification



La méthode notoire « Waterfall »



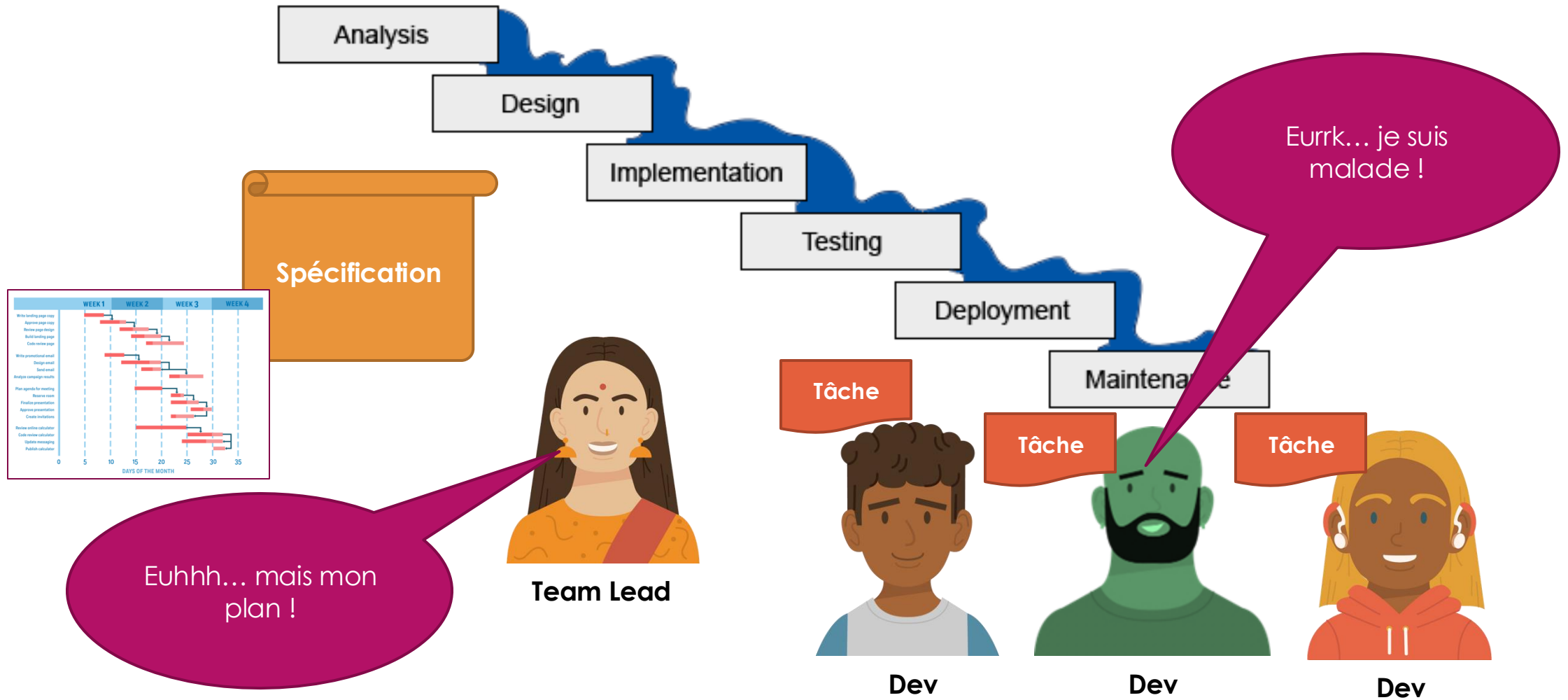
La méthode notoire « Waterfall »

- ▶ Le gérant du projet doit tout savoir en avance
 - ▶ Toutes les dépendances entre les tâches
 - ▶ Le temps nécessaire pour faire une tâche

C'est quoi une dépendance ?

Qui estime le temps ? Et comment ?

La méthode notoire « Waterfall »



Des imprévus

- ▶ Pendant un projet, il y a **toujours** des imprévus
 - ▶ On tombe malade
 - ▶ Des coéquipiers partent de l'entreprise
 - ▶ On découvre une nouvelle dépendance ou contrainte
 - ▶ Un détail caché et seulement révélé pendant l'exécution
 - ▶ Une modification hors notre contrôle ? (règlement, politique)
- ▶ Comment adapter ?



Team Lead

J'ai assuré à mon chef que tout sera prêt en 2 ans !



CEO

On m'a promis un retour sur l'investissement, et je l'ai promis à mes investisseurs !



Utilisateur

On m'a dit que j'aurai ma nouvelle appli !

Désolé les gars, on
va travailler les soirs
et weekends



Team Lead



Dev



Dev

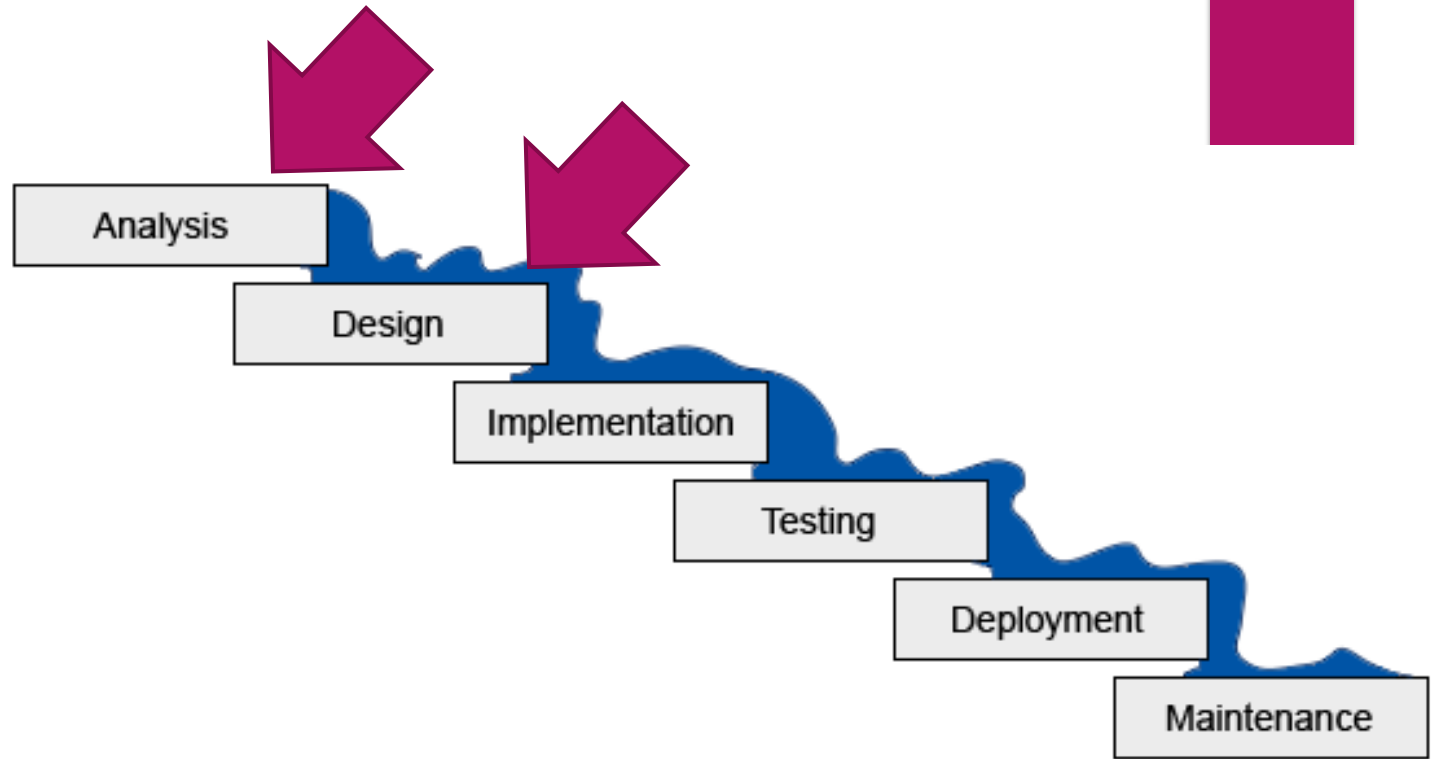


Dev

On fait marche
arrière et on
adapte la spec et
le plan



Team Lead

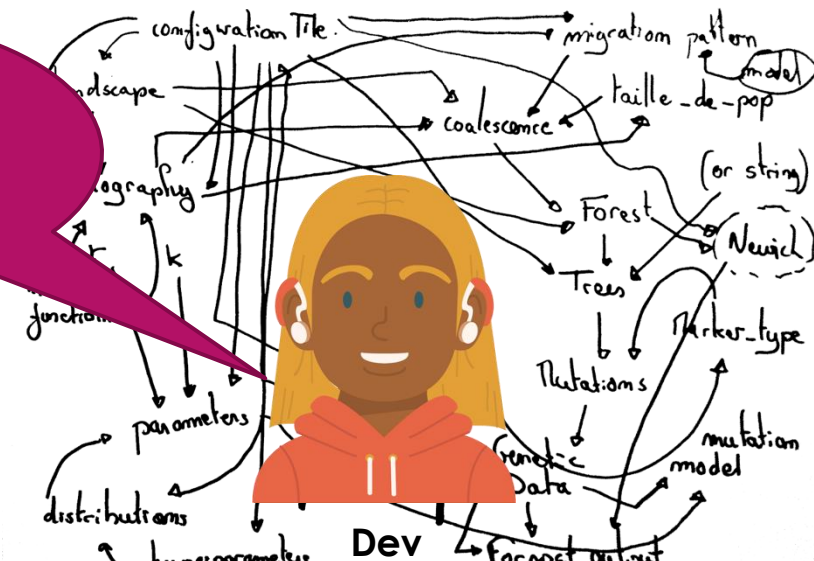


Bon, à la poubelle
tout mon travail....
C'est la vie

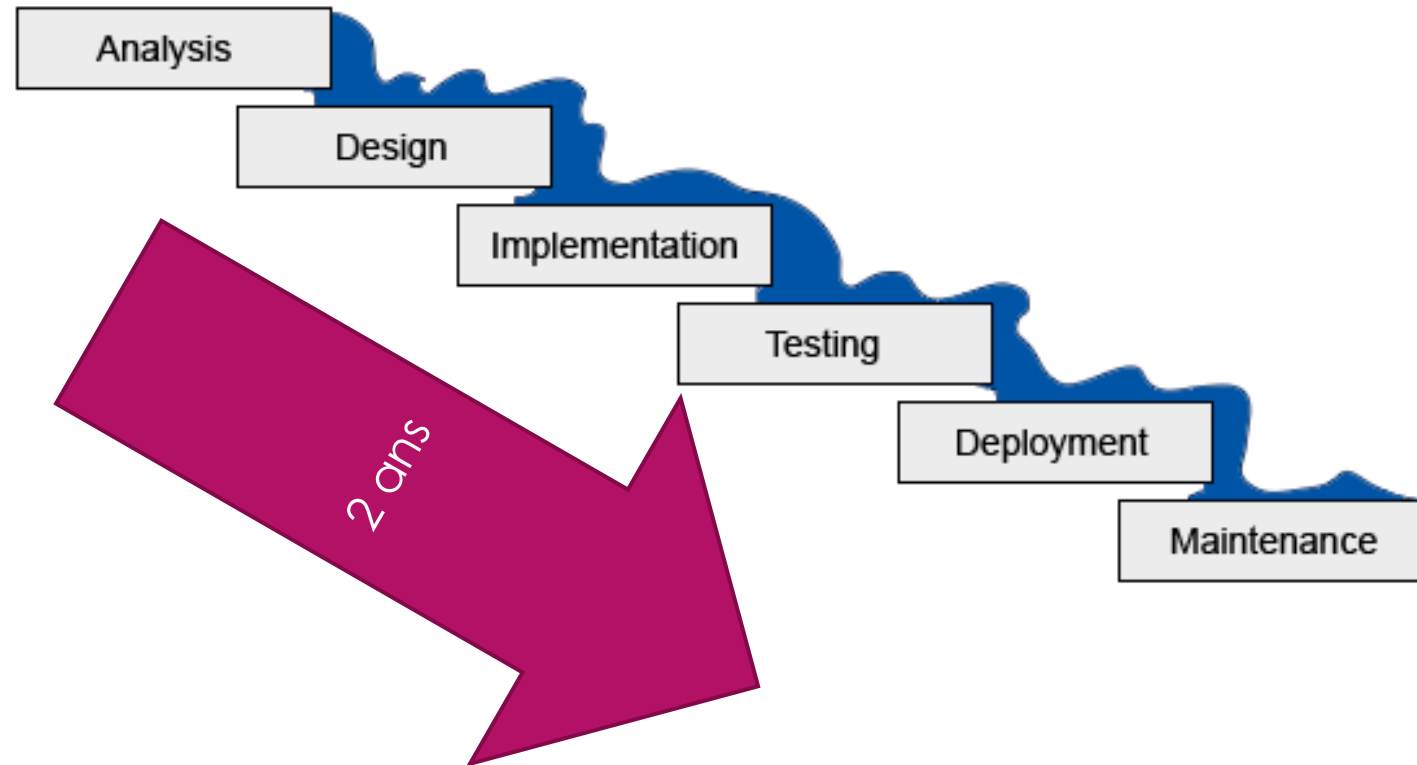


Dev

On n'a pas le
temps de tout
refaire, je vais
adapter mon
code existant



La méthode notoire « Waterfall »



C'est fini !!!!

Logiciel

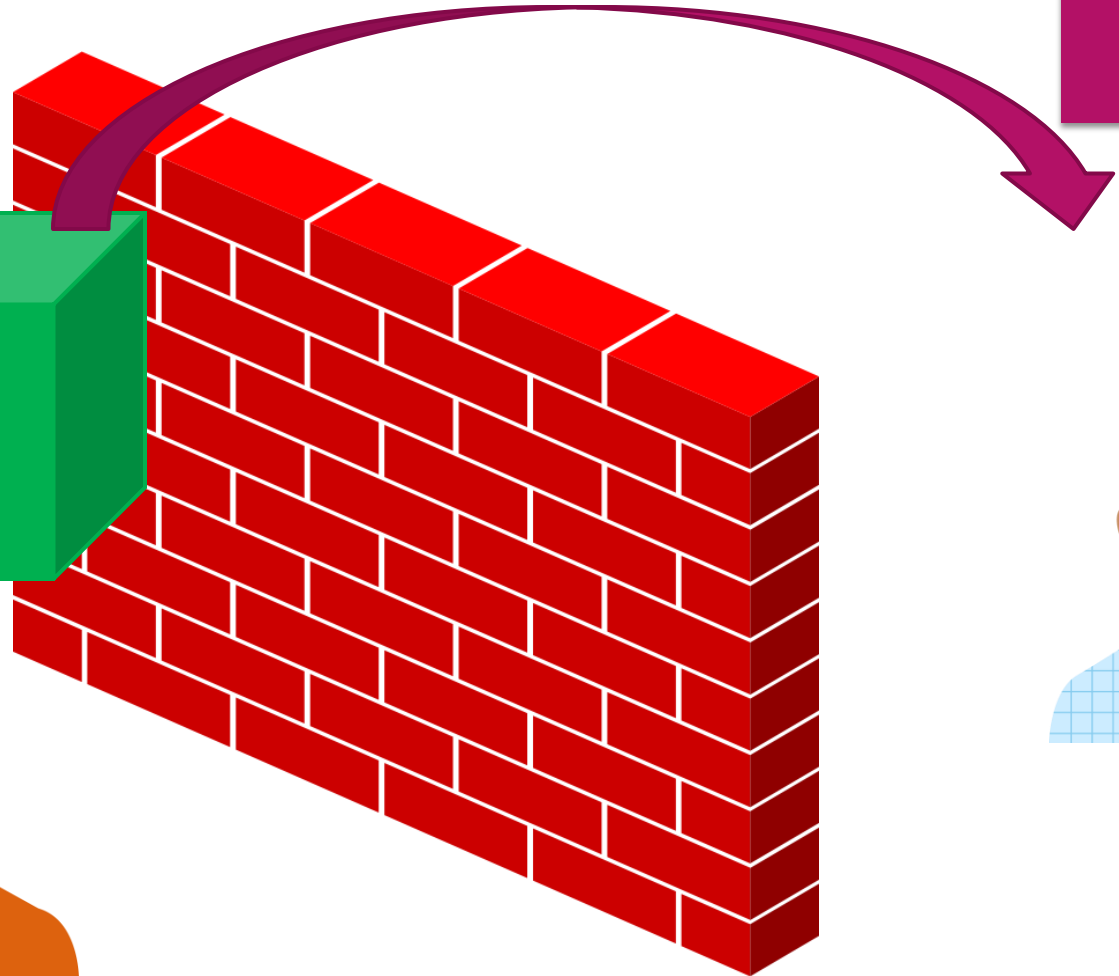
Utilisateur

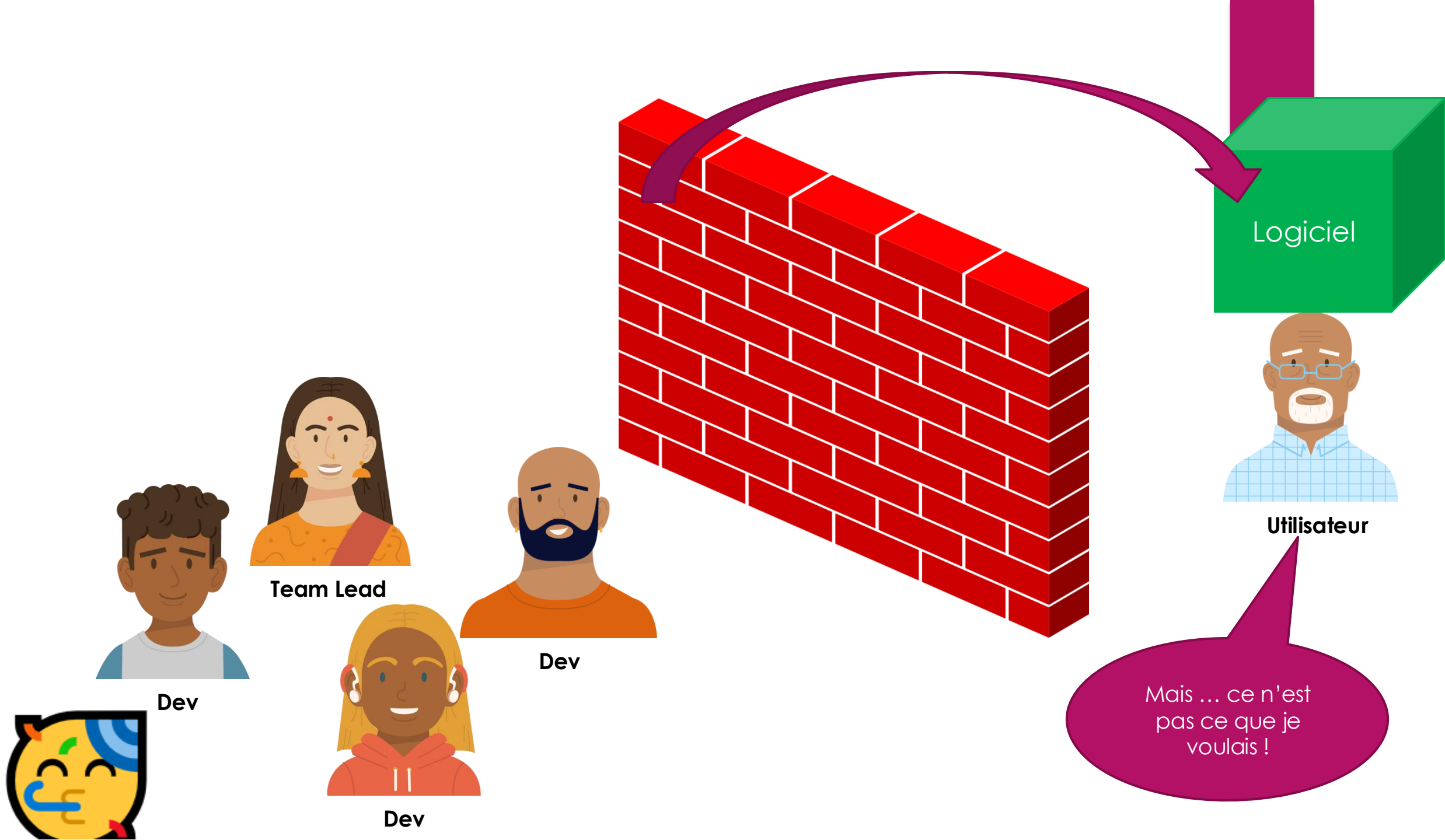
Team Lead

Dev

Dev

Dev





La méthode notoire « Waterfall »

- ▶ Pendant le 2 ans (ou plus) d'exécution

- ▶ Les besoins ont changés
- ▶ Le monde a changé

Comment le monde peut changer ? Pensez à l'exemple de la **VR !**

Comment cela peut se faire ?

Est-ce qu'on est certain des besoins initiaux ?

- ▶ Logiciel instable

- ▶ Bugs créés à cause des modifications
- ▶ Code « spaghetti » difficile à maintenir

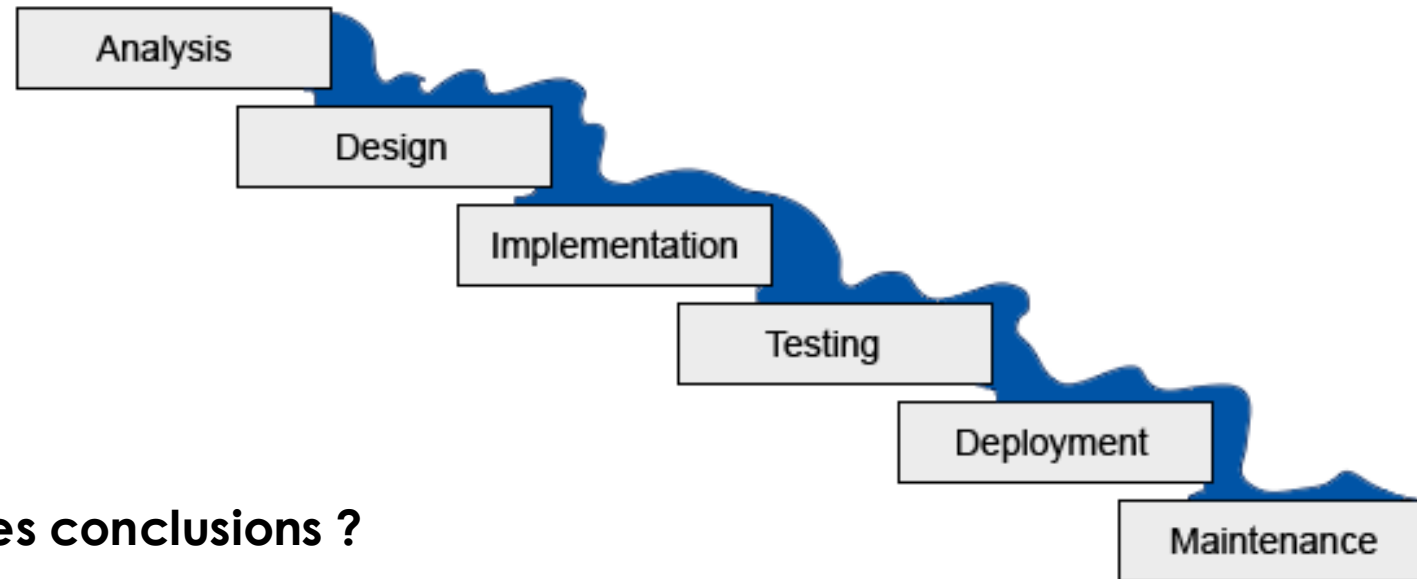


CEO



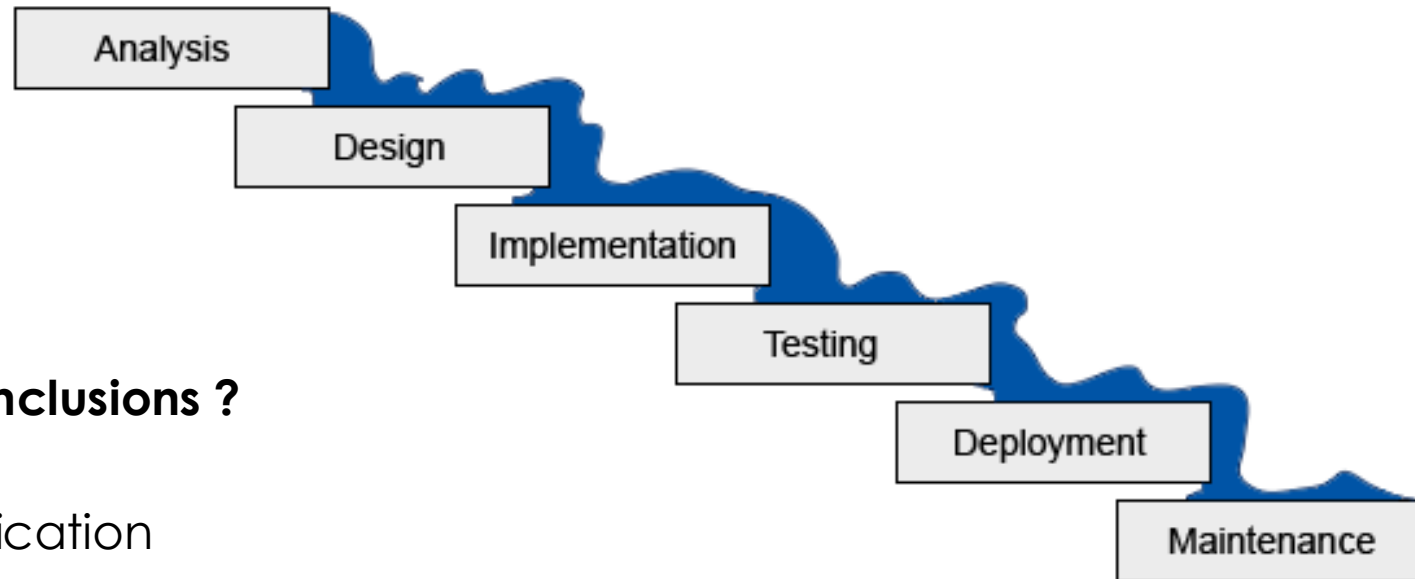
Utilisateur

La méthode notoire « Waterfall »



Avez-vous des conclusions ?

La méthode notoire « Waterfall »



Avez-vous des conclusions ?

Communication

Flexibilité

Adhérence aux méthodes

Documentation

AGILE



J'ai entendu parler d'agile...

”

RACONTEZ MOI VOTRE IDÉE D'AGILE !

Agile peut dire plusieurs choses selon le rôle



**Project
Manager**

On peut plus
facilement gérer
des modifications

Chaque rôle a sa propre
vision d'agile et
l'implémente

Résultat :

« *Better than not doing it* »

(*mieux que rien faire*)

Une réunion tous
les jours ;
Task-board ;
Burndown charts



Team Lead

Des outils de test et
automatisation



Dev

Une perspective fracturée ?



Une perspective fracturée ?



Utilisateur



**Project
Manager**

User story :

Lecteur de musique pour un
bar doit jouer de préférence
les dernières tubes



Dev

Une perspective fracturée ?



Utilisateur

Je vais attirer un public plus jeune dans mon bar



Project Manager

Je vais pouvoir communiquer à notre client notre prochain livrable

User story :

Lecteur de musique pour un bar doit jouer de préférence les dernières tubes



Team Lead

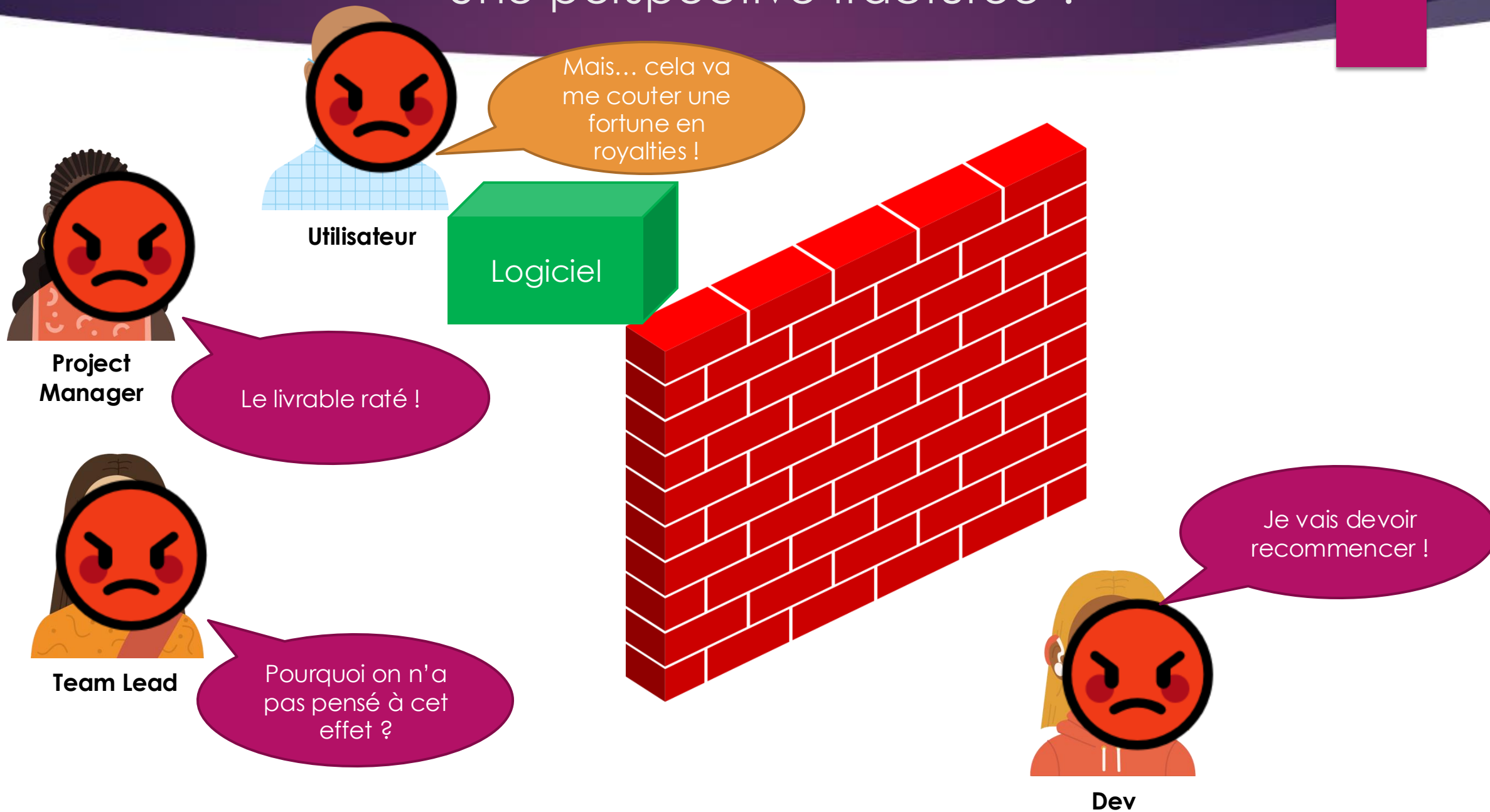
C'est le prochain objectif à planifier

Implémentation : toujours sélectionner les derniers tubes



Dev

Une perspective fracturée ?



« Agile Manifesto »

- AGILE est avant tout un **système de valeurs**

Nous découvrons de meilleures façons de développer des solutions,
par notre propre pratique et en aidant les autres dans leur pratique.

Grâce à ce travail, nous en sommes venus à valoriser :

Les individus et leurs interactions, de préférence aux processus et aux outils,
Des solutions opérationnelles, de préférence à une documentation exhaustive,
La collaboration avec les clients, de préférence aux négociations contractuelles,
La réponse au changement, de préférence au respect d'un plan.

Précisément, même si les éléments à droite ont de la valeur,
nous reconnaissons davantage de valeur dans les éléments à gauche.

*Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler,
James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick,
Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*

« Agile Manifesto »

- AGILE est avant tout un système de valeurs

Nous découvrons de meilleures façons
de travailler par notre propre pratique et en aidant les autres.

Grâce à ce travail, nous en

Les individus et leurs interactions, de préférence aux processus et aux outils,
Des solutions opérationnelles, de préférence à une documentation exhaustive,
La collaboration avec les clients, de préférence aux négociations contractuelles,
La réponse au changement, de préférence au respect d'un plan.

Précisément, même si les éléments à droite ont de la valeur,
nous reconnaissons davantage de valeur dans les éléments à gauche.

On aurait pu éviter des problèmes de nos exemples si il y a eu des meilleurs communications entre l'équipe et le client :
plus souvent, plus efficace et en personne

*Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler,
James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick,
Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*

« Agile Manifesto »

- AGILE est avant tout un **système de valeurs**

Nous découvrons de meilleures façons de développer des solutions,
par notre propre pratique et en aidant les autres dans leur pratique.

Grâce à ce travail, nous en sommes venus à valoriser :

Les individus et leurs interactions, de préférence aux processus et aux outils,
Des solutions opérationnelles, de préférence à un contrat négocié,
La collaboration avec les clients, de préférence à la négociation,
La réponse au changement, de préférence à suivre un plan.

Si on avait montré au fur et à mesure notre team lead et ou client, on aurait pu détecter plus tôt des problèmes.

Précisément, même si les éléments à valoriser sont les mêmes,
nous reconnaissons davantage de valeur dans les individus et leurs interactions

*Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler,
James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick,
Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*

« Agile Manifesto »

- AGILE est avant tout un système de valeurs

Nous découvrons de meilleures façons de développer des logiciels
par notre propre pratique et en aidant les autres dans la leur.

Grâce à ce travail, nous en sommes capables.

Les individus et leurs interactions, de préférence aux processus et aux outils,
Des solutions opérationnelles, de préférence aux négociations de contrat,
La collaboration avec les clients, de préférence aux négociations de contrat,
La réponse au changement, de préférence au respect d'un plan.

Collaboration régulière, des retours clients, l'agilité de changer le projet avec des besoins.

Adhérer à un contrat ou imposer de la rigidité assure des mauvais résultats

Précisément, même si les éléments à droite ont de la valeur,
nous reconnaissons davantage de valeur dans les éléments à gauche.

*Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler,
James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick,
Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*

« Agile Manifesto »

- AGILE est avant tout un systeme de valeurs

Nous découvrons de meilleures façons de développer des solutions,
par notre propre pratique et en aidant les autres dans leur pratique.

Grâce à ce travail, nous en sommes venus à valoriser :

Les individus et leurs interactions, de préférence aux processus et aux outils,
Des solutions opérationnelles, de préférence à une documentation exhaustive,
La collaboration avec les clients, de préférence aux négociations contractuelles,
La réponse au changement, de préférence au respect d'un plan.

Précisément, même si les éléments à d
nous reconnaissons davantage de valeur da

Trop souvent on adhère au
« plan » fait par ou approuvé par
des autres pour se dégager de
notre responsabilité.

*Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham,
James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian M.
Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*

Dans le doute... Retourner au manifeste !

- ▶ Il n'est pas simple d'adopter ou mettre en pratique les valeurs Agiles
- ▶ Pour cela
 - ▶ On les décortique en 12 directives pour mieux les comprendre
 - ▶ On a développé un nombre de méthodologies qui aident à leur adoption
 - ▶ Scrum
 - ▶ On a développé un nombre d'outils qui permet de plus facilement mettre en pratique les valeurs
 - ▶ User Stories
 - ▶ Timeboxing
 - ▶ Task boards

Si la méthodologie ne fonctionne pas,
Ou un outil est plus contraignant que
bénéficient
Alors ...on les abandonne en faveur
des individus et les interactions !

Mais, n'oubliez jamais la première ligne du manifeste !!!

Les individus et leurs interactions, de préférence aux processus et aux outils,

Les ennemis de l'Agile



**Commande
et contrôle**

Le CEO, le gérant du projet, le team-lead

... qui soi-disant « adopte » les valeurs Agiles

... en impose une méthodologie (ex. scrum) ou des outils (ex. daily standup)

... mais seulement pour mieux « contrôler » le projet

**Délègue toujours des tâches quotidiens
pendant le Daily Standup**

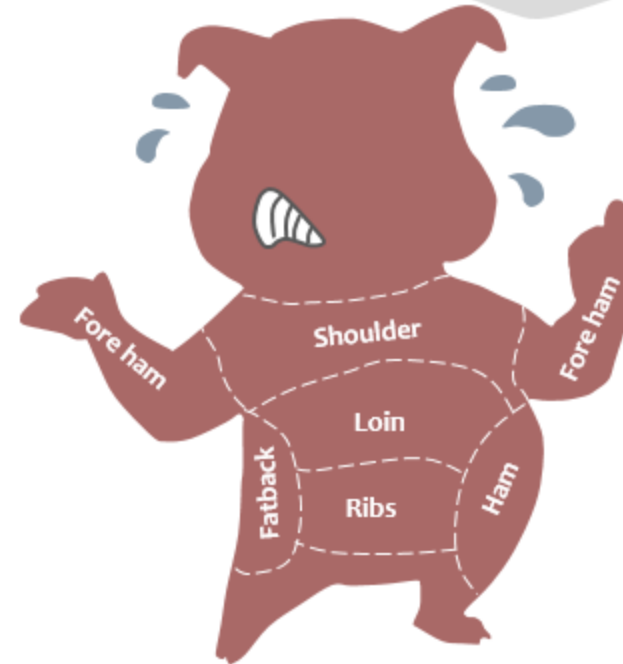
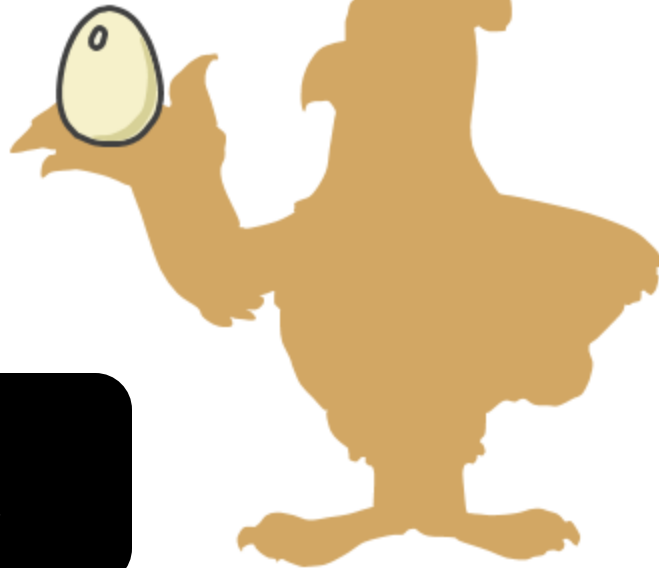
N'écoute pas l'équipe

Résistant au changements

L'individualiste



Involvement vs **Commitment**



Mentalité CYA
« cover your ass »

L'individu « pro » vs « personnel »

- ▶ Les valeurs Agiles ne vous demandent pas de tout sacrifier au projet
- ▶ Nous avons tous des priorités personnelles
 - ▶ famille, vie privée, repose, passe-temps, projets persos
 - ▶ **tout est valable et à respecter !**
 - ▶ L'adoption réussie de valeurs Agiles finira par pouvoir bien respecter ces priorités !
- ▶ Il y a aussi des priorités professionnelles
 - ▶ Satisfaction, qualité de travail, augmentation, promotion
 - ▶ Parfois **une priorité professionnel pourrait contredire** des valeurs Agiles
 - ▶ Ex. j'aimerais apprendre techno X, et même si personne d'autre dans l'équipe le connais, je vais l'imposer pour ce projet

Il n'est pas facile de distinguer. Mais poser la question :
est-ce que c'est bon pour le collectif (y compris le client) ?

Un logiciel fonctionnel servira mieux à avancer votre
carrière qu'une faillite !

Les 12 principes

1. Prioriser la satisfaction du client

« **Early delivery** »
(livrer tôt)

« **Continuous delivery** »
(livrer souvent)

Fait ce que je
dis... pas ce que
j'ai dit !

Le client ne sais
ce qu'il veut

Perceptions
différentes de la
valeur

Les 12 principes

2. Les changements sont bienvenus !

...même tard dans le projet

Personne n'est coupable

On est tous impliqué, et unis !

On n'attend pas, on est réactif sur le moment

Un changement n'est pas une bêtise

On apprend !

Difficile de gérer
ses émotions

Respect

Fierté (orgueil ?)

Les 12 principes

3. Livrer en permanence des versions opérationnelles

« Timeboxing »

**Donner l'occasion à tous les partis prenants
de « se rendre compte »**

Constater tôt les changements !

Planning
compressé

Commande et
contrôle ?



Demos !

Les 12 principes

4. La communication la plus efficace est la conversation en personne

Construire un sens de communauté

Pouvoir s'exprimer des bonnes et mauvaises nouvelles

Perceptions
variées

Communication par chat
ou réseaux sociaux
souvent pas « filtrée »

« Busy work »

Les 12 principes

5. L'équipe « business » et l'équipe « dev » doivent travailler ensemble !

Comprendre et apprécier les enjeux des autres !

... permet de détecter tôt les problèmes et anticiper

Elargir son point
de vu

Apprécier ce qu'il a le
plus de valeur... Où
mettre les priorités

Les 12 principes

6. Construire les projets autour des personnes motivées

Leur faire confiance !

Chacun se sent responsable et a une notion de possession du projet

Mentalité CYA

« Busy work »
« Micro-management »

Les 12 principes

7. La mesure principale de progrès et le produit opérationnel

Pouvoir tester et utiliser le produit et la meilleure manière de communiquer son état de développement

Perceptions
différentes du
progrès

Je suis à 95% ...

Les 12 principes

8. Rythme constant et soutenable

Estimations réalistes

**Une équipe débordée livre essentiellement de
mauvais produits**

« burnout » est
mauvais pour
tout le monde

Les 12 principes

9. Attention continue à la qualité et l'excellence technique

Prenez le temps à réfléchir
(ou accorder ce temps à vos coéquipiers)

**Utilisez des bonnes pratiques développés par des autres (ex.
design patterns, architectures, frameworks, outils, etc)**

« Le prix s'oublie,
la qualité reste »

« If you think good
design is expensive,
you should look at the
cost of bad design »

Les 12 principes

10. Simplicité ... maximiser le quantité de travail à ne pas faire

Réduire des dépendances

Eviter la complexité

Documentation

Réunions

« Busy work »

Planning

Les 12 principes

11. Les meilleurs architectures et produits s'émergent des équipes auto-organisantes

Design incrémental

Meilleure maîtrise
des problèmes

Responsabilité
partagée

Les 12 principes

12. Réflexion régulière sur comment s'améliorer

On assume les problèmes vécus

On cherche à apprendre

Honnêteté

Parfois ce n'est
pas très
confortable

Les 12 principes

Satisfaire le client

Accueillir les changements

Livrer régulièrement un produit
fonctionnel

Communication face à face

Equipes qui se composent des
personnes « business » et « dev »

Personnes motivés

Un produit fonctionnel est la
mesure de progrès

Procès soutenable

Attention à la qualité

Simplicité

Equipe auto-organisantes

Réflexion et amélioration



L'Agile en pratique : SCRUM

Une méthodologie et boîte à outils

Philosophie vs Vie Quotidienne

- ▶ Il est facile de **dire** qu'on va adopter les valeurs Agiles
- ▶ Il est plus compliqué de les **mettre en pratique**

Je ne partage
pas la même
vision

C'est une perte
de temps

Où est mon
plan ?



Philosophie vs Vie Quotidienne

- ▶ Les adoptants d'Agile ont inventé (par expérience durement acquise) des **méthodologies** afin de nous aider et guider
 - ▶ Une façon de penser
 - ▶ Une désignation de rôles
 - ▶ Un set d'outils et méthodes
 - ▶
- ▶ Une de ses **méthodologies** s'appelle « SCRUM »

Attention !

Nous allons apprendre le vocabulaire SCRUM ainsi que les méthodes et outils...

MAIS

N'oubliez jamais le 1^{er} point du manifeste Agile

« **Les individus et leurs interactions**, de préférence aux processus et aux outils »

Rôles



Utilisateur



CEO

J'ai assez de responsabilité pour prendre de décisions concernant le produit final

Je travail en liaison avec des parties prenants (utilisateurs, clients, ...)

Je gère des attentes des clients

Je maintien la liste des priorités



Product Owner



Scrum Master

SCRUM



Dev



Dev



Dev

Une hiérarchie plate et équitable

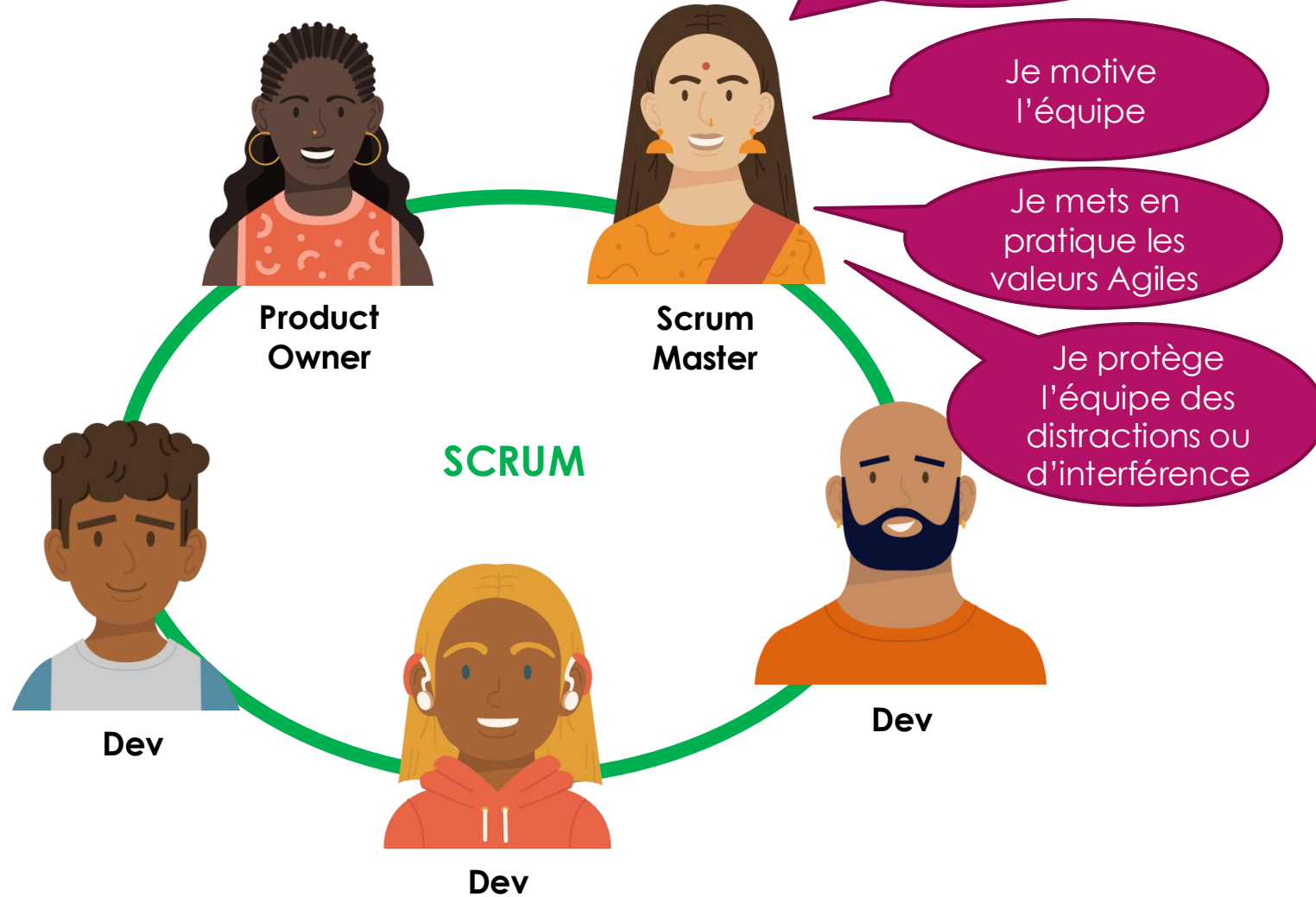
Rôles



Utilisateur



CEO



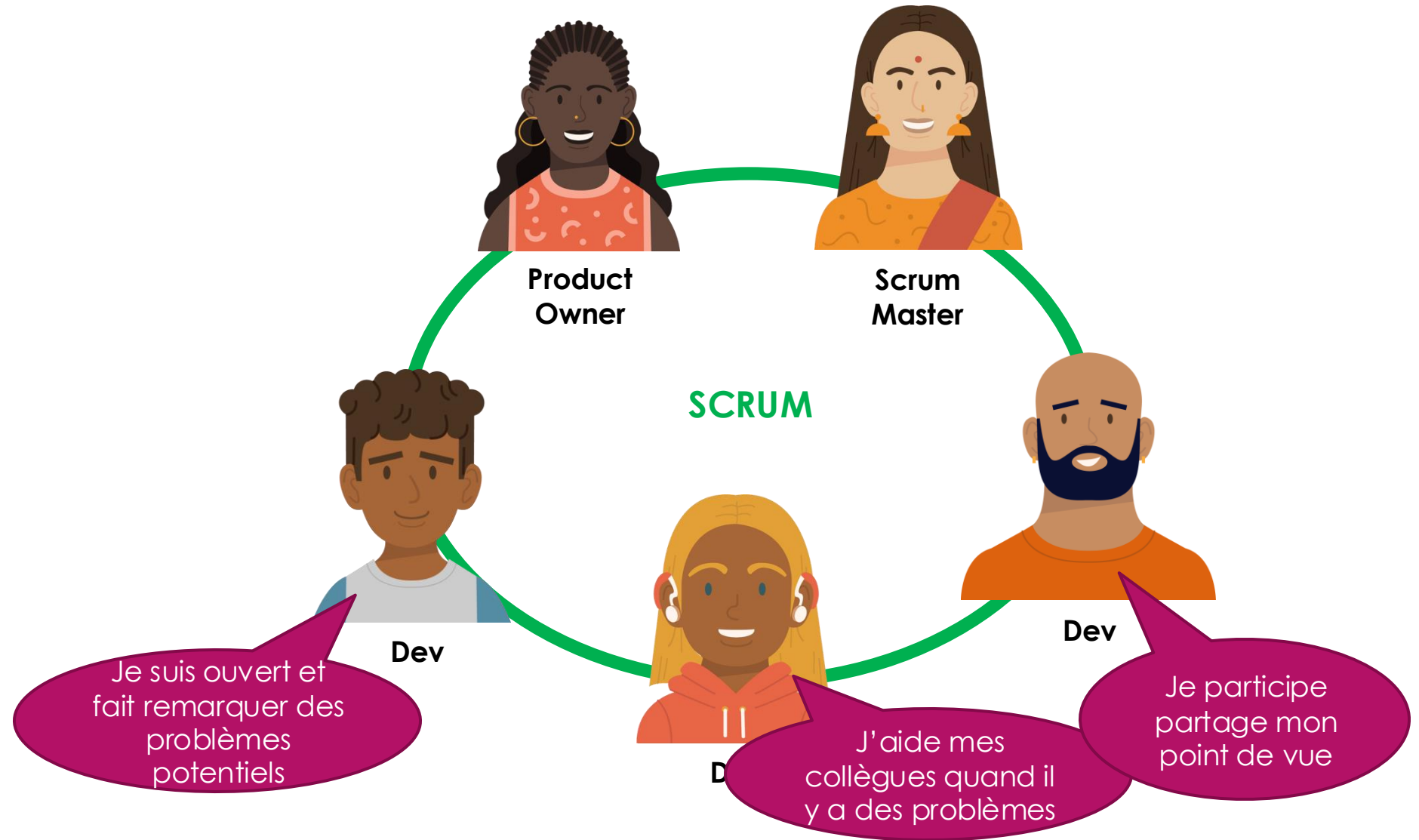
Rôles



Utilisateur



CEO



Outil 1 : le « product backlog »

- ▶ La liste de **demandes** des clients
- ▶ La liste de **features** à venir
- ▶ La liste de **problèmes** ou bugs à corriger

Product Backlog

Pouvoir émettre une facture

Le client devrait rassembler la liste d'ateliers animés pendant la période et sortir une facture

Rendre l'interface accessible pour les non-voyants

Nos clients non-voyants ont du mal à naviguer l'interface avec les outils comme VoiceOver.

Souvent juste une collection de fiches bostols collées au tableau blanc



Product Owner



Utilisateur



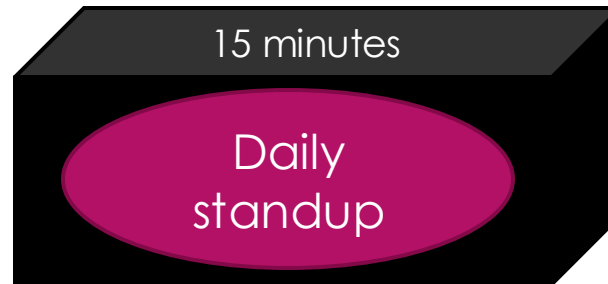
CEO

discussion

Je discute avec les partis prenants, et je rédige, maintient la liste « à faire ». Je les **priorise** constamment !

Outil 2 : le « Timebox »

- ▶ Toutes les activités sont limitées en temps
 - ▶ Tout le monde est au courant de la limitation
 - ▶ Tout le monde s'engage à cette limite
 - ▶ Surtout : personne ne promet rien qui risque de dépasser la limite !

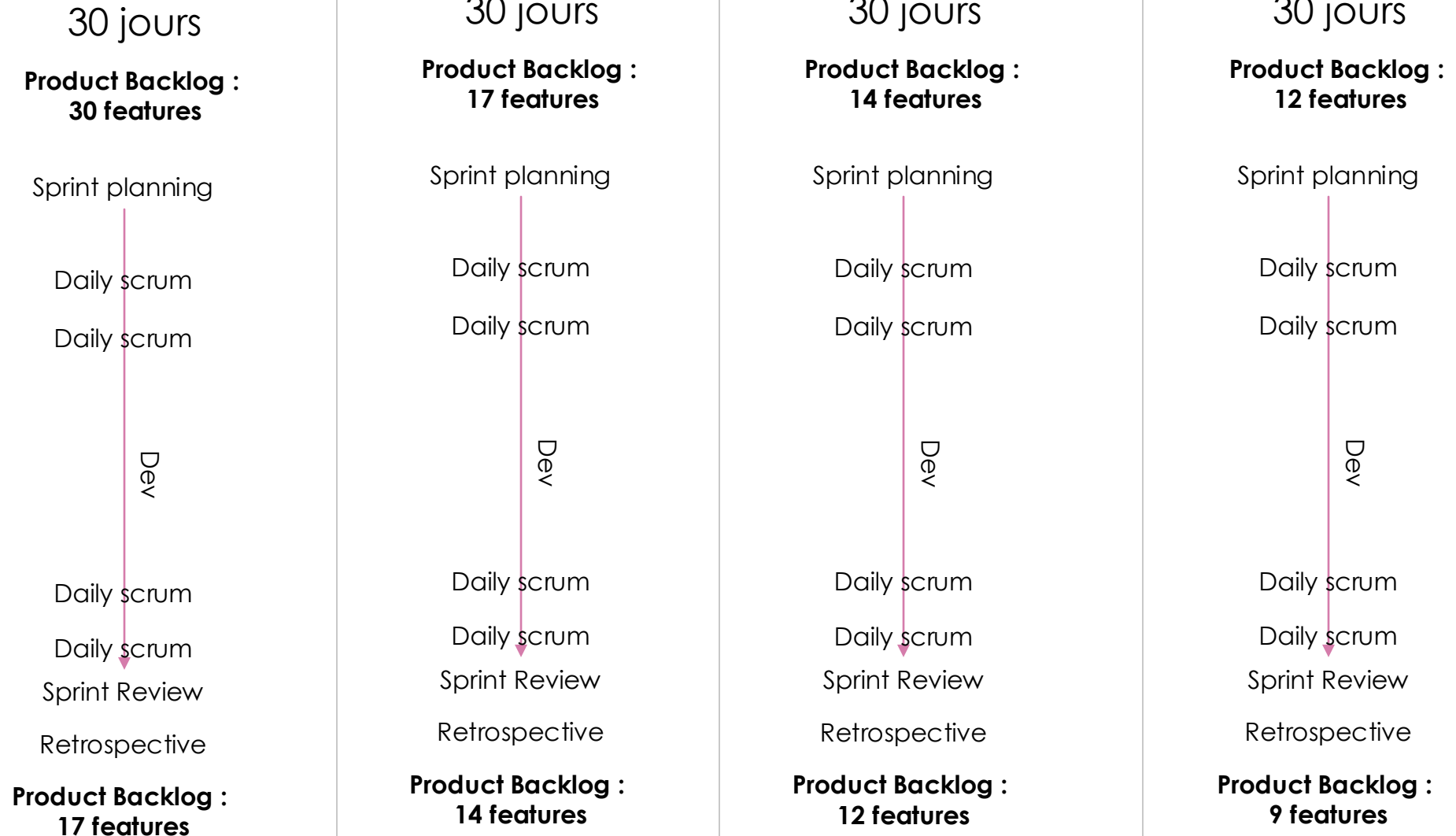


Outil (méthode) 3 : le « **sprint** »

- ▶ Un **sprint** est une itération de développement fait par l'équipe
 - ▶ L'équipe décide ensemble les features du **Product Backlog** à implémenter
 - ▶ Surtout : l'équipe ne promet uniquement ce qu'elle décide collectivement est faisable et raisonnable
- ▶ Un sprint est aussi limité en temps : souvent 1 mois, ou 2 semaines
- ▶ Les objectifs du sprint sont décomposés en sous-tâches, l'ensemble qui s'appelle le **Sprint Backlog**.
- ▶ Au fur et à mesure, l'équipe traite, ajoute, enlève des tâches du **Sprint Backlog**
- ▶ A la fin de la période :
 - ▶ Un produit fonctionnelle est sorti, et présenté lors du **Sprint Review**
 - ▶ L'équipe fait une **retrospective** afin de s'améliorer

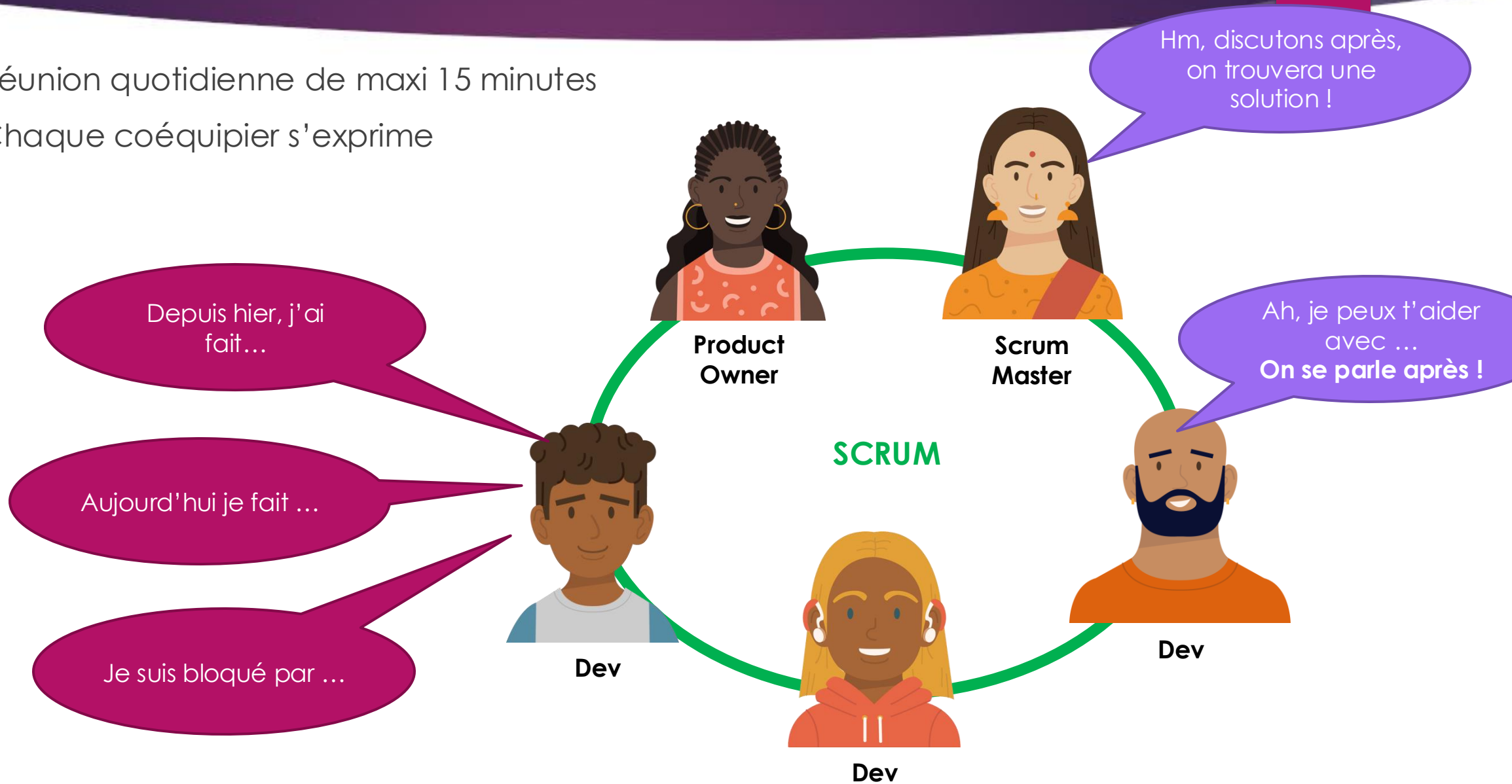


Les sprints



Outil 4: le « Daily Scrum »

- Réunion quotidienne de maxi 15 minutes
- Chaque coéquipier s'exprime



Outil 5: Le « Task Board »

Sprint Backlog

A faire

Pouvoir émettre une facture

Le client devrait rassembler la liste d'ateliers animés pendant la période et sortir une facture

Implémenter
Interface
front

Tests
automatiques

En cours

Implémenter
les APIs
Julien

Renderer du
document PDF

Sylvie

Fait

Modifier le
schéma de
la base de
données

Outil 6 : le « sprint review »



Utilisateur



CEO

Nous vous avons
promis un module de
facturation... voici
une démo de ce
module

Pouvoir émettre une facture

Le client devrait rassembler la
liste d'ateliers animés pendant
la période et sortir une facture

Live Demo



Product
Owner



Scrum
Master



Dev



Dev



Dev

Outil 6 : le « sprint review »



Utilisateur

Super, j'aurais aimé
plutôt faire comme
ça



CEO

En fait le règlement vient
tout juste de changer, il
faudrait qu'on émette la
facture en format



Product
Owner

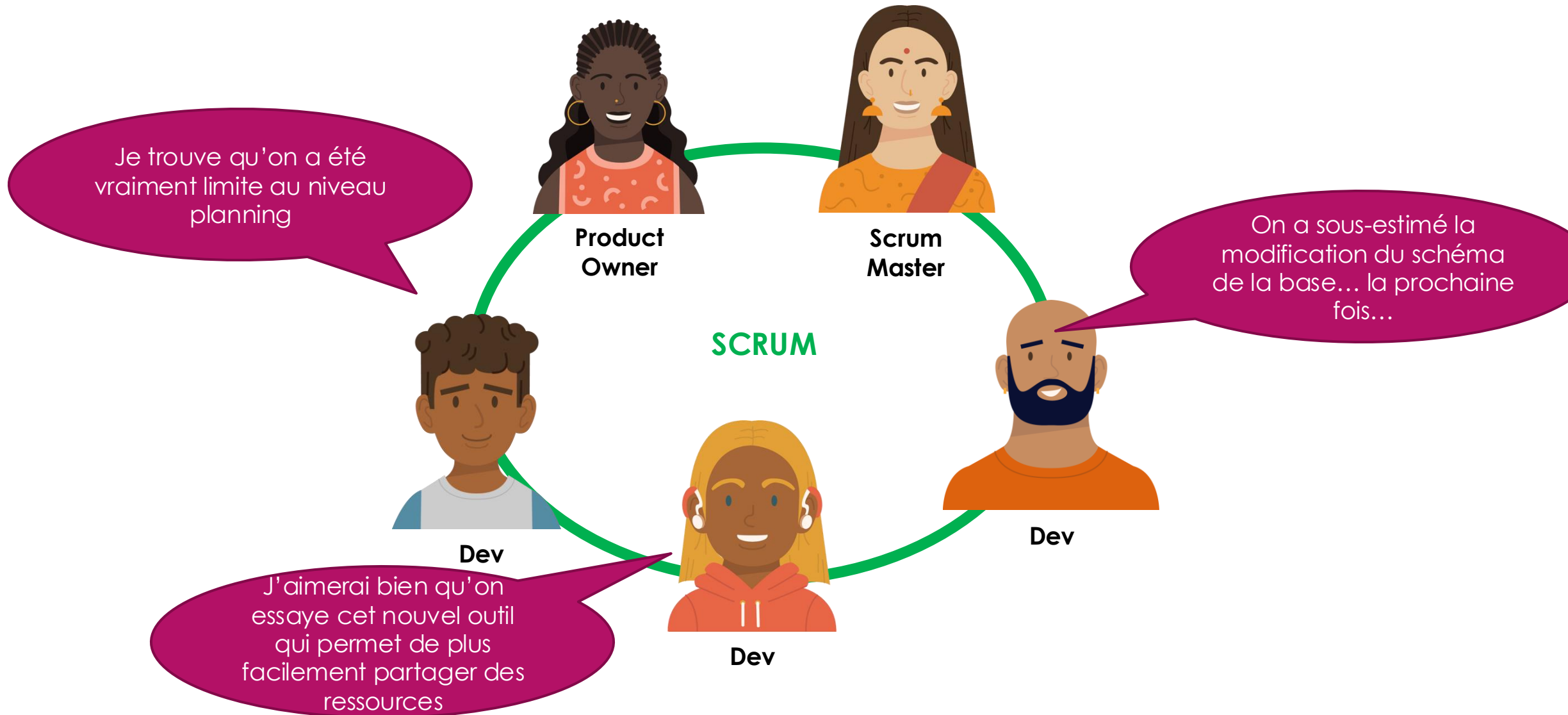
Product Backlog

Modifier l'ergonomie

Lirom epsom

Ajouter un format X

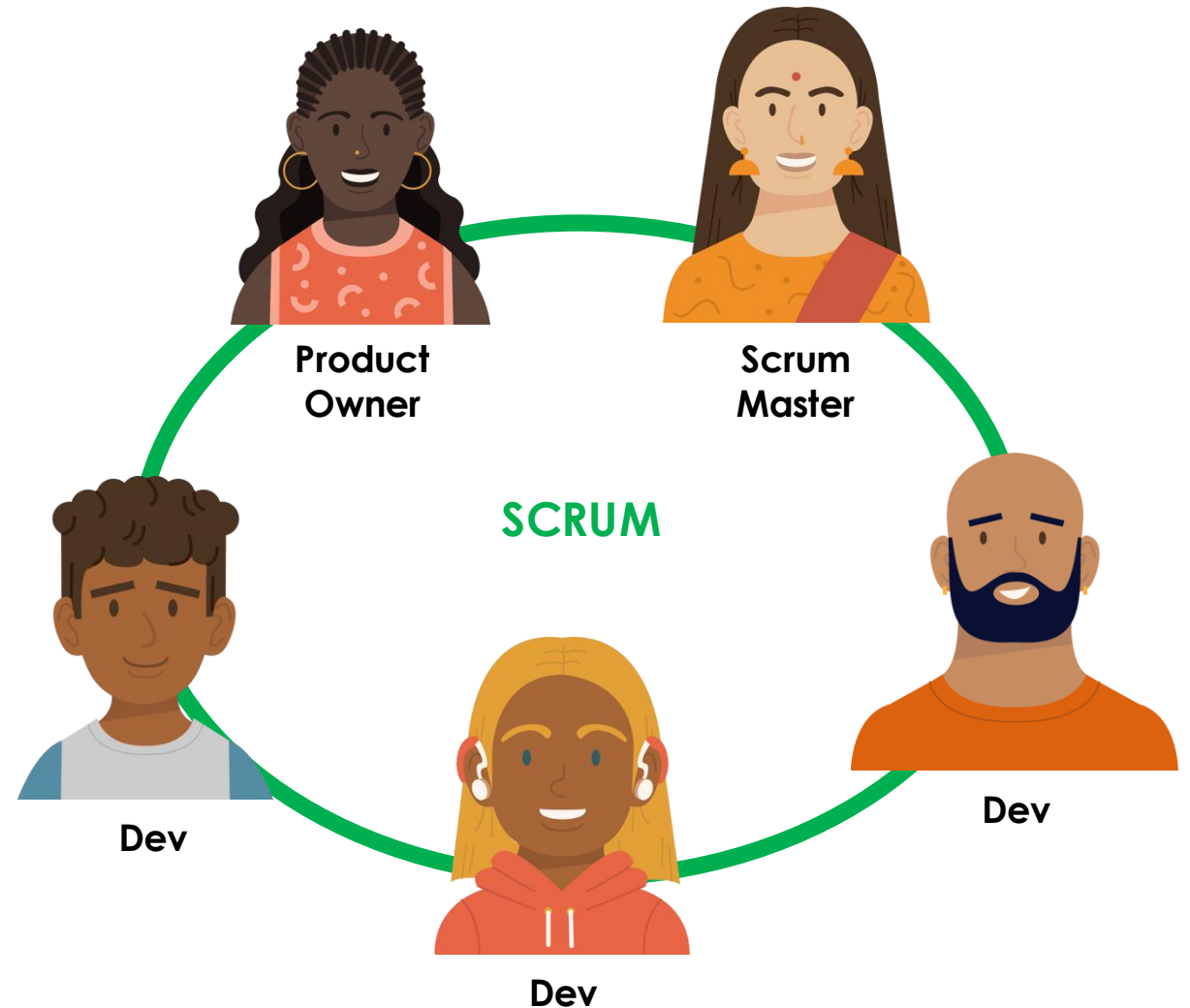
Outil 7 : la « retrospective »



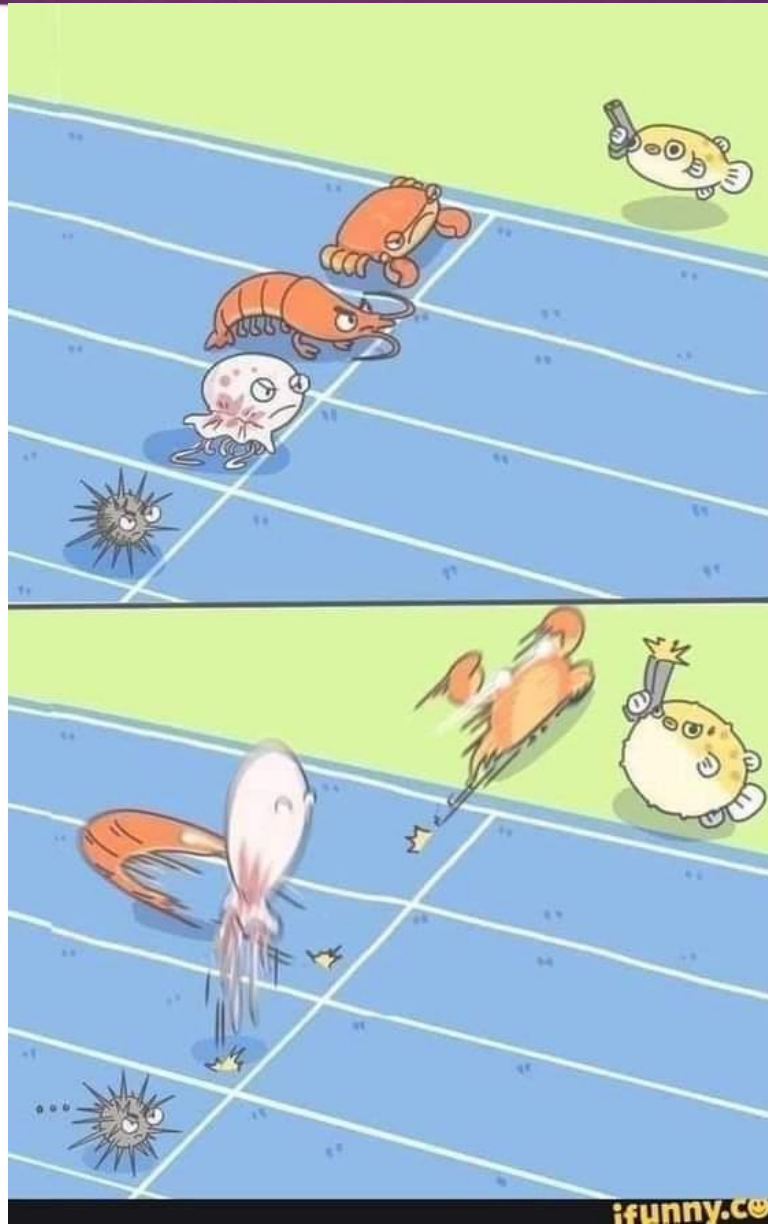
Sprint planning

- ▶ Tout le monde de l'équipe est **propriétaire du plan**
- ▶ Chacun s'engage
- ▶ Engagement collectif

Mais comment développer cet engagement ??



Sprint planning



Sprint planning

► Chacun de l'équipe se croit au plan !

► Comment ?

► Outil : le **User Story**

Après discussion avec les utilisateurs, je résume les besoins en tant que **User Story**



Product Owner

Pouvoir émettre une facture

En tant que comptable
je veux créer une facture des ateliers du mois
afin de gérer et suivre plus facilement les paiements clients

Qui ?
Quoi ? Que faire ?
Pourquoi ?



Dev



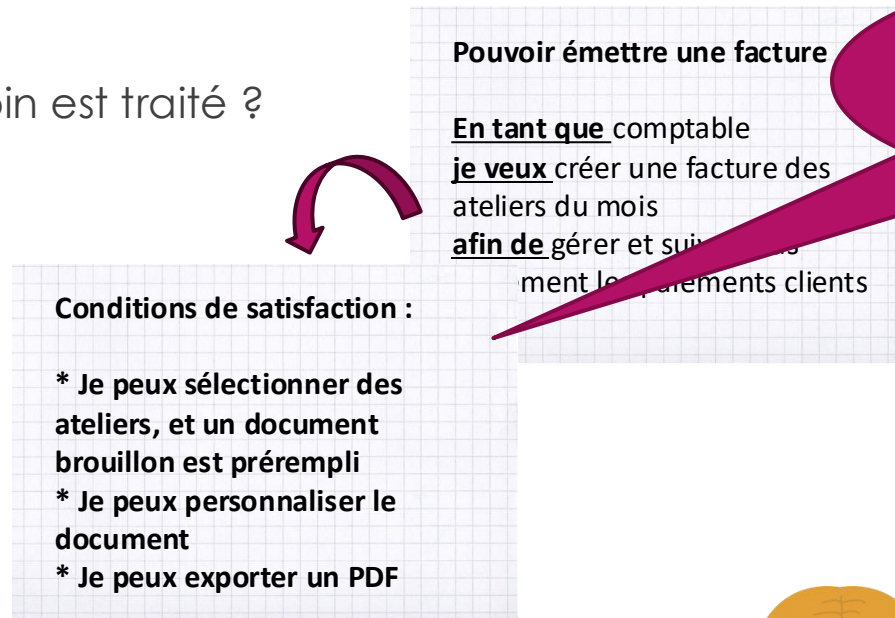
Dev

D'accord, on comprends !



Sprint planning

- ▶ Commente limite de la compréhension **divergente** ou **variée** ?
- ▶ Outil: **conditions de satisfaction**
 - ▶ Comment on évalue que le besoin est traité ?
 - ▶ Dans le produit opérationnel
 - ▶ Point 2 du manifeste agile !



La tâche est considéré comme « **fait** » uniquement si pendant la démo je peux effectuer toutes ces actions



Dev



Dev



Ah c'est plus clair !

Sprint planning

- Mais comment estimer si le User Story est réalisable en un seul sprint ?



Dev

Ben ... je ne sait pas trop combien de temps ça me prendra



Dev

Conditions de satisfaction :

- * Je peux sélectionner des ateliers, et un document brouillon est prérempli
- * Je peux personnaliser le document
- * Je peux exporter un PDF

Pouvoir émettre une facture

En tant que comptable
je veux créer une facture des ateliers du mois
afin de gérer et suivre plus ment les paiements clients

Discutons ensemble !



Scrum Master



Dev

Le feature est grand... ça me semble irréalisable

Sprint planning

► La séance de **Sprint Planning**

- Max 4h pour discuter et choisir les User Stories à mettre dans le sprint
- Max 4h pour décortiquer les User Stories et créer le Sprint Backlog

Pouvoir émettre une facture

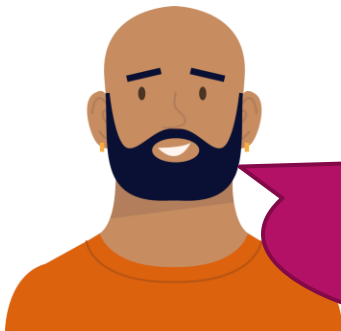
En tant que comptable

je veux créer une facture des ateliers du mois

afin de gérer et suivre plus facilement les paiements clients

Module de rendu
d'un PDF

Modifier le schéma
de la base de
données



Dev

D'abord on va devoir
créer un module
capable de sortir un
fichier PDF



Dev

On aura besoin de
modifier le schéma de
la base de données



Dev

Sprint planning

► La séance de **Sprint Planning**

- Discussion entre l'équipe aide à détecter de contraintes ou problèmes
- ... qui révèle d'autre tâches à ajouter

Pouvoir émettre une facture

En tant que comptable

je veux créer une facture des ateliers du mois

afin de gérer et suivre plus facilement les paiements clients

Module de rendu
d'un PDF

Modifier le schéma
de la base de
données

Ecrire des
migrations

Oui, mais si on modifie le
schéma de la base de
données, il faut mettre à
jour les données existantes !



Dev



Dev

Ah ! Bien vu !
**On devrait penser à
écrire de migrations !**



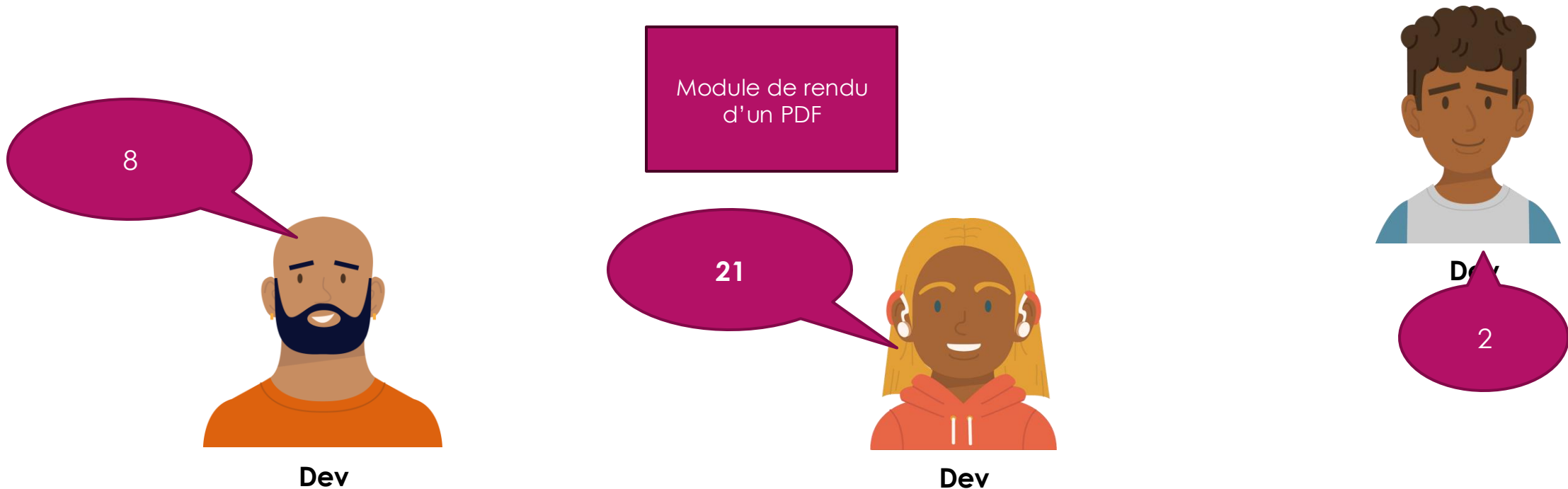
Dev

Sprint planning

- ▶ L'équipe est d'accord sur les sous tâches... mais combien de temps ? Est-ce que tout est réalisable en un seul sprint ?

- ▶ **Story points**

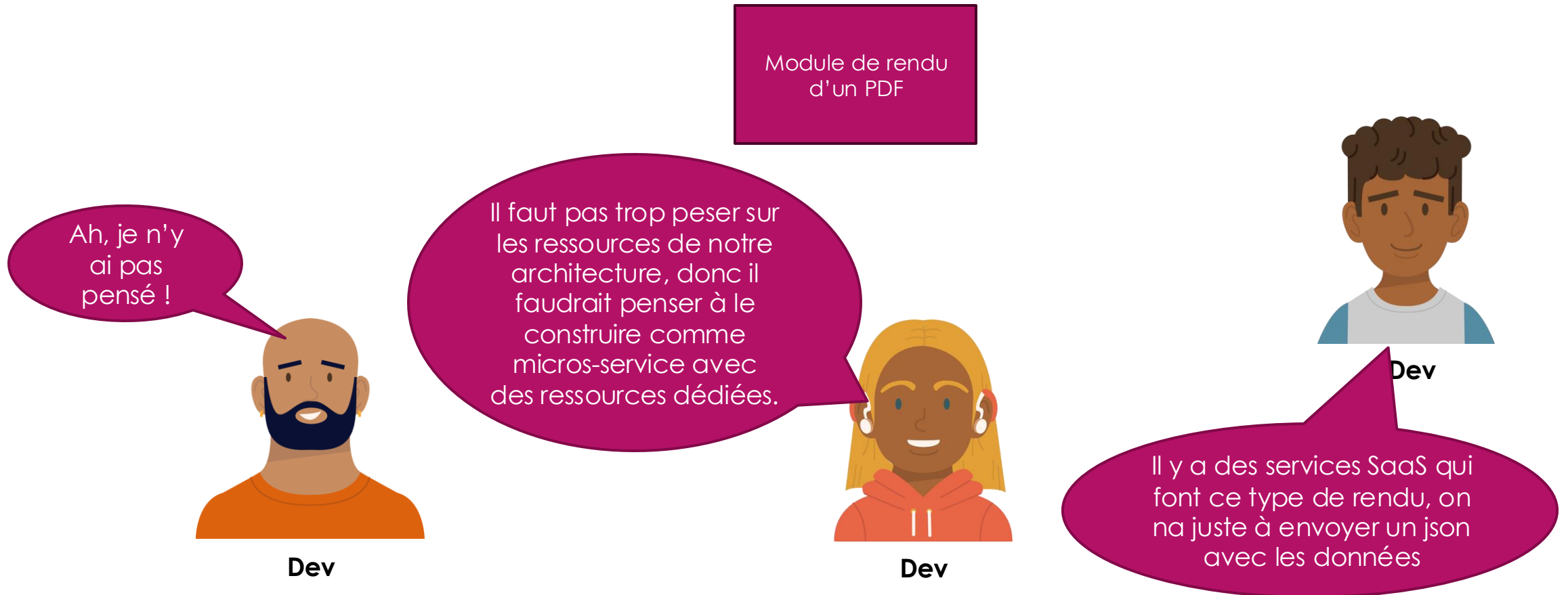
- ▶ On attribue des points à chaque tâche
- ▶ « **Story point poker** » : chaque membre écrit sur une carte une valeur du set: 1, 2, 3, 5, 8, 13, 21 (séquence Fibonacci)... plus la valeur est élevée, plus c'est longue / difficile



Sprint planning

- ▶ Story point poker :

- ▶ Ensuite, s'il y a une divergence, on explique (sans culpabilité) notre raisonnement



Sprint planning

On s'est tous investi dans la construction du plan !
On est propriétaire !
On est engagé !

► Tout le monde se met d'accord

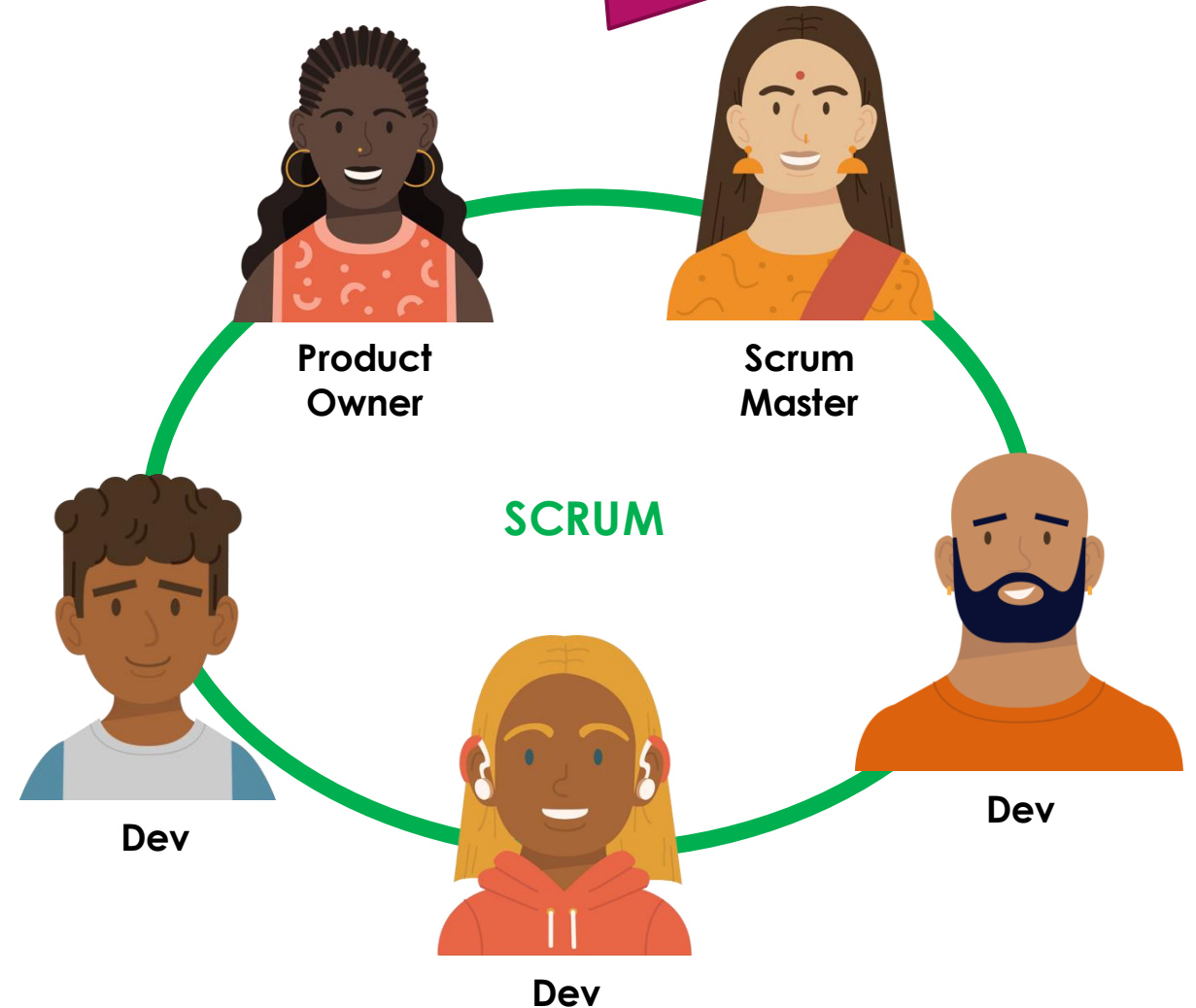
Pouvoir émettre une facture

En tant que comptable
je veux créer une facture des ateliers du mois
afin de gérer et suivre plus facilement les paiements clients

Module de rendu
d'un PDF
(16)

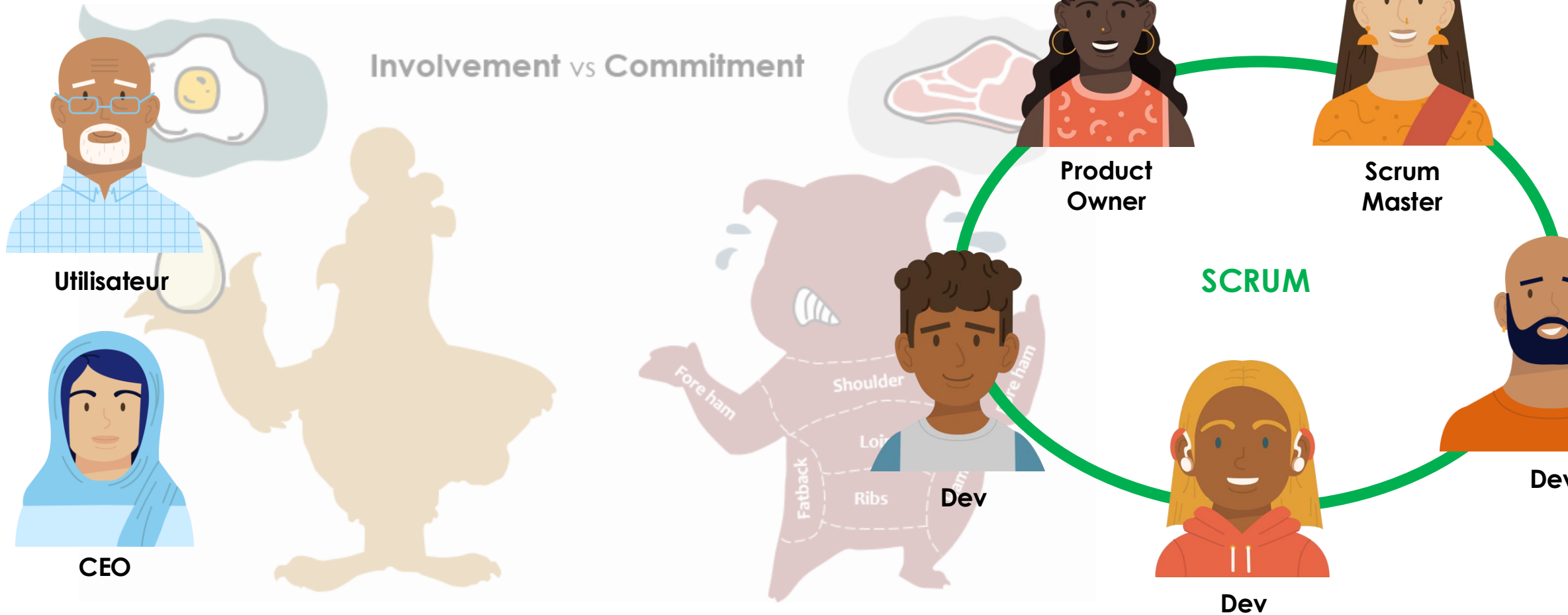
Modifier le schéma
de la base de
données
(5)

Ecrire des
migrations
(4)



Sprint planning

- ▶ Le sprint planning ne fonctionne pas si
 - ▶ un ou plusieurs membre de l'équipe ont la philosophie « poule » (ou CYA)
 - ▶ le scrum master ou gérant est de type « commande et contrôle »



Les valeurs de SCRUM

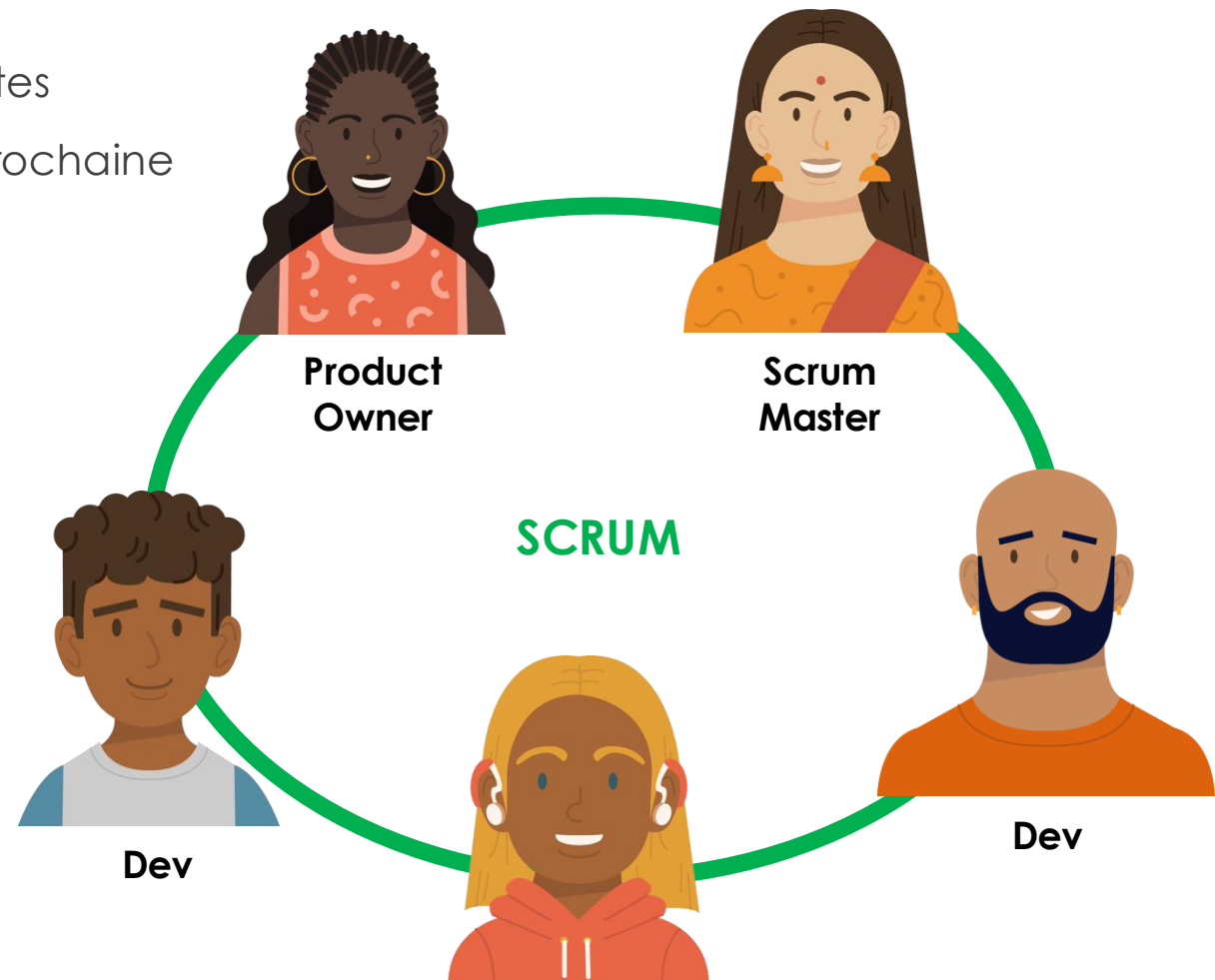
- ▶ Chaque coéquipier est **engagé**
- ▶ Du **respect** entre coéquipiers
- ▶ **Focus !**
- ▶ Esprit **ouvert**
- ▶ **Courage**

« Daily Scrum »

► La réunion quotidien **courte**

- **Inspection quotidien** de tout l'équipe (devs et Product Owner) du projet
- Facilement détecter des dérapages
- Facilement détecter des problèmes ou contraintes
- Chaque coéquipier sélectionne soi-même sa prochaine tâche du sprint backlog
- Prise de décision **au plus tard possible**

On a tous une vision et une compréhension commune du projet



Prise de décision au plus tard possible

- ▶ ... en restant raisonnable
- ▶ Il est très commun de se créer des tâches trop grandes ou complexes
- ▶ En exécutant la tâche on découvre sa complexité
 - ▶ Pas grave, on la décompose en sous-tâche et on les ajoute au Sprint Backlog

« Given enough eyeballs, all bugs are shallow » - Linus Torvalds

Prise de décision au plus tard possible

Sprint Backlog

A faire

Pouvoir émettre une facture

Le client devrait rassembler la liste d'ateliers animés pendant la période et sortir une facture

Implémenter
Interface
front

Tests
automatique
s

Ecrire des
migrations

En cours

Implémenter
les APIs
Julien

Renderer du
document PDF
Sylvie

Fait

Modifier le
schéma de
la base de
données

Je pense que je doit écrire en urgence les migrations avant les tests automatiques, car ça risque de le bloquer plus tard



Dev

Daily Scrum

- ▶ « Daily Standup »
 - ▶ Quelques praticiens préfèrent que tout le monde soit debout, pour assurer que la réunion soit courte !
- ▶ Règles de réussite :
 - ▶ Chacun est un cochon !
 - ▶ Les conversations détaillées sont pour plus tard
 - ▶ Tous les jours, quelqu'un différent commence
 - ▶ Ce n'est pas un rituel
 - ▶ Tout le monde participe
 - ▶ Ce n'est pas un « progress report »
 - ▶ Inspecter toutes les tâches, et celles à venir
 - ▶ Changer le plan si nécessaire !

L'exécution d'un sprint

- ▶ Le Product Owner est très impliqué dans le succès d'un sprint
 - ▶ Il **élève** les objectives
 - ▶ Il encourage tout l'équipe à parler de la **valeur** qu'ils sont en train de livrer

Motivation pour l'équipe !

Creuser un trou ?

Creuser un trou pour
protéger mon village
d'une inondation

L'exécution d'un sprint

- ▶ Vous vous souvenez des Sprint Score ?
- ▶ On peut l'utiliser pour suivre le progrès de notre sprint avec un **Burndown Chart**

Sprint Backlog

A faire

Pouvoir émettre une facture

Le client devrait rassembler la liste d'ateliers animés pendant la période et sortir une facture

Implémenter
Interface
front
(13)

Tests
automatiques
(8)

Ecrire des
migrations
(4)

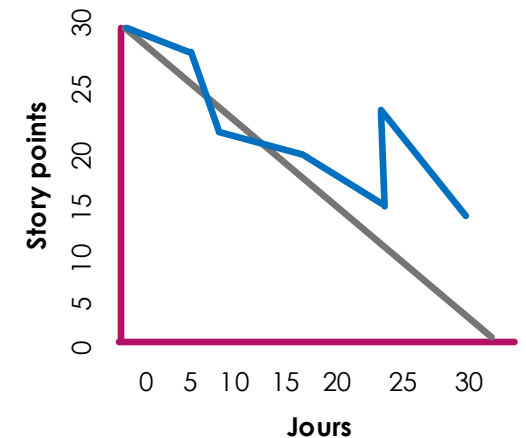
En cours

Implémenter
les APIs
(13)
Julien

Renderer du
document PDF
(16)
Sylvie

Fait

Modifier le
schéma de la
base de
données
(4)



Le tour de Victoir

- ▶ Le sprint est terminé
- ▶ Un démo réussi a été fait
- ▶ C'est le temps de revoir ce qui s'est passé pendant le dernier sprint et voir les points d'amélioration
 - ▶ La prochaine fois on va évaluer plus précisément des Sprint Points
 - ▶ On ne va pas oublier le point X, Y, Z
 - ▶ Nos Daily Scrums était trop longues
 - ▶ Etc...

Avertissement

- ▶ Dans l'industrie il y a beaucoup de personnes et entreprises qui proclament adopter Agile
 - ▶ Parce qu'ils font une réunion quotidienne qui s'appelle le « Daily Scrum »
 - ▶ Sauf la réunion se déroule en mode « commande et contrôle » où le gérant distribue des tâches de la journée
 - ▶ Parce qu'ils maintiennent un « Task board » avec plein de petits post-its
 - ▶ Sauf en réalité l'équipe n'a pas de mot sur l'affectation des tâches ou une compréhension de leur valeur
 - ▶ L'équipe travaille en mode « poule », et se dispense de la responsabilité
 - ▶ Parce qu'on développe en « itération » et qu'on les appelle des « sprint »
 - ▶ Mais on ne laisse pas tester le client jusqu'à la toute fin du projet
- ▶ L'utilisation d'une **méthode** ou un **outil** Agile / Scrum ne veut pas dire que vous êtes Agile !

Être réellement Agile veut dire l'adoption des valeurs !

Avertissement

- ▶ Employer les **méthodes** ou **outils** sans adopter les **valeurs** pourrait avoir un effet net-positif
 - ▶ **Mais on aura seulement des résultats « mieux que sans rien faire »**
 - ▶ ... toujours avec de échéances ratées, un équipe fatiguées, des produits inutilisables
- ▶ Ceci engendre une déception de tout le monde qui attendait de la « *performance et productivité époustouflante* »
- ▶ ... qui fini par **l'abandon** de l'Agile

Vous êtes poule ou cochon ?

