

Scrum

**Formation aux pratiques de la
méthode agile la plus populaire**



Support de cours

2012



aubryConseil

Claude Aubry

Auteur

- Blog Scrum, agilité et rock'n roll

~~Professeur~~

- Module agile de l'UF ISI
- Projets tutorés en Scrum

Coach

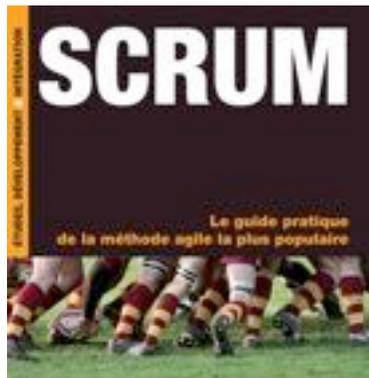
- Coach agile en entreprise

Open Source

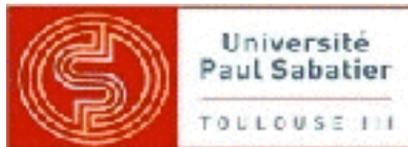
- Product Owner d'iceScrum
- Associé Kagilum

Communauté

- SigmaT
- Fédération Agile
- SUG



aubryConseil



www.sigmat.fr

Kagilum

Fédération
Agile
Francophone

Cette formation est animée par un membre de la Fédération Agile

- Regroupement de formateurs et coaches
 - Avec une expertise reconnue dans les méthodes agiles
 - Impliqués dans les communautés francophones
 - Venant d'horizons différents mais partageant les mêmes valeurs
 - charte en ligne : www.federationagile.com
- Vos bénéficiaires
 - Le support de cours est revu régulièrement par les pairs
 - Les pratiques présentées sont actualisées en commun
 - La pédagogie est enrichie grâce au feedback de tous
 - Le formateur est aussi un coach qui pratique

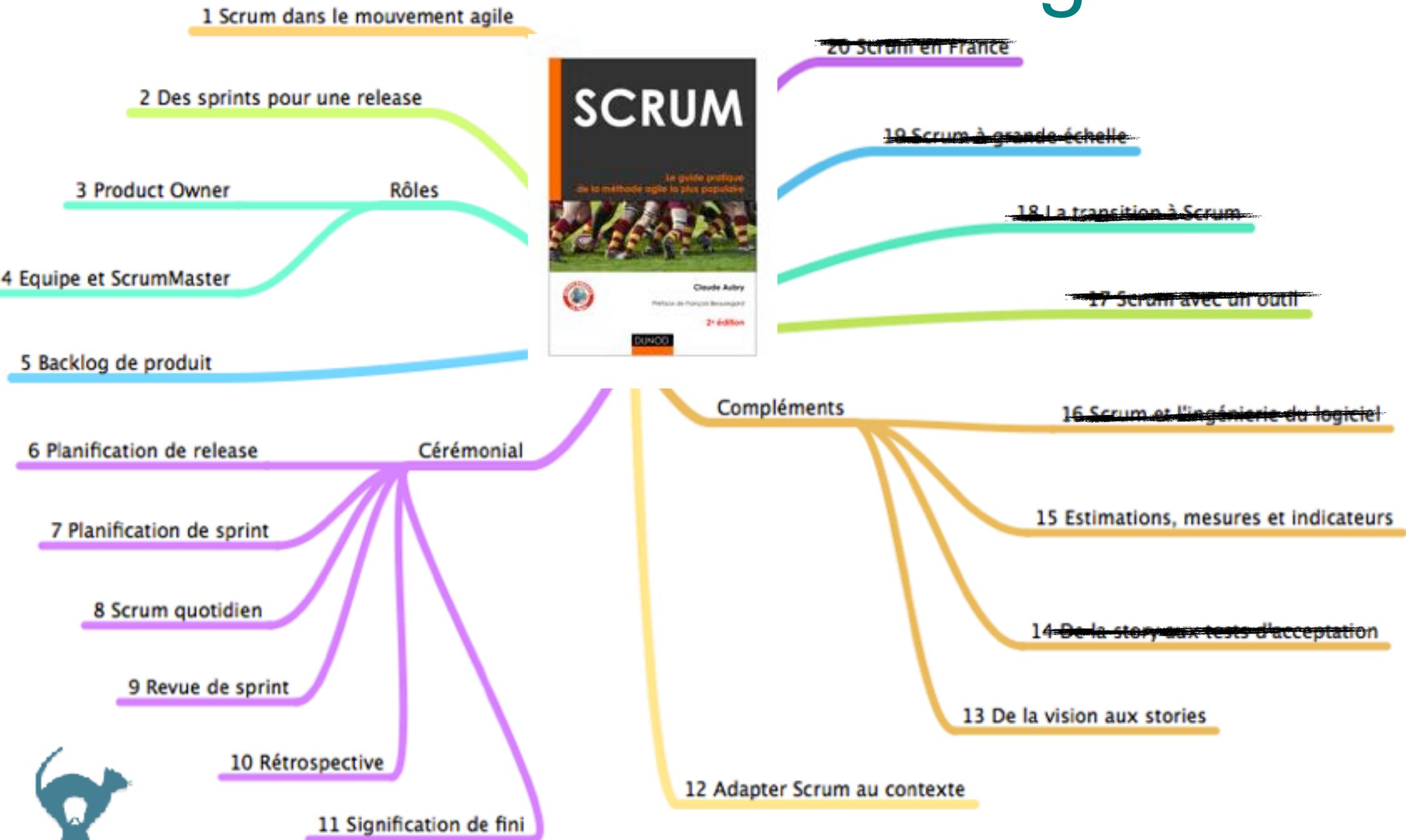


Objectifs

- ▶ Scrum fournit un cadre simple pour une équipe qui travaille sur un projet, avec un cérémonial précis et limité.
- ▶ Grâce à des exposés, des discussions, des exercices et l'examen de vos problèmes particuliers, vous apprendrez comment appliquer Scrum sur vos projets et l'adapter quand c'est nécessaire.



Programme

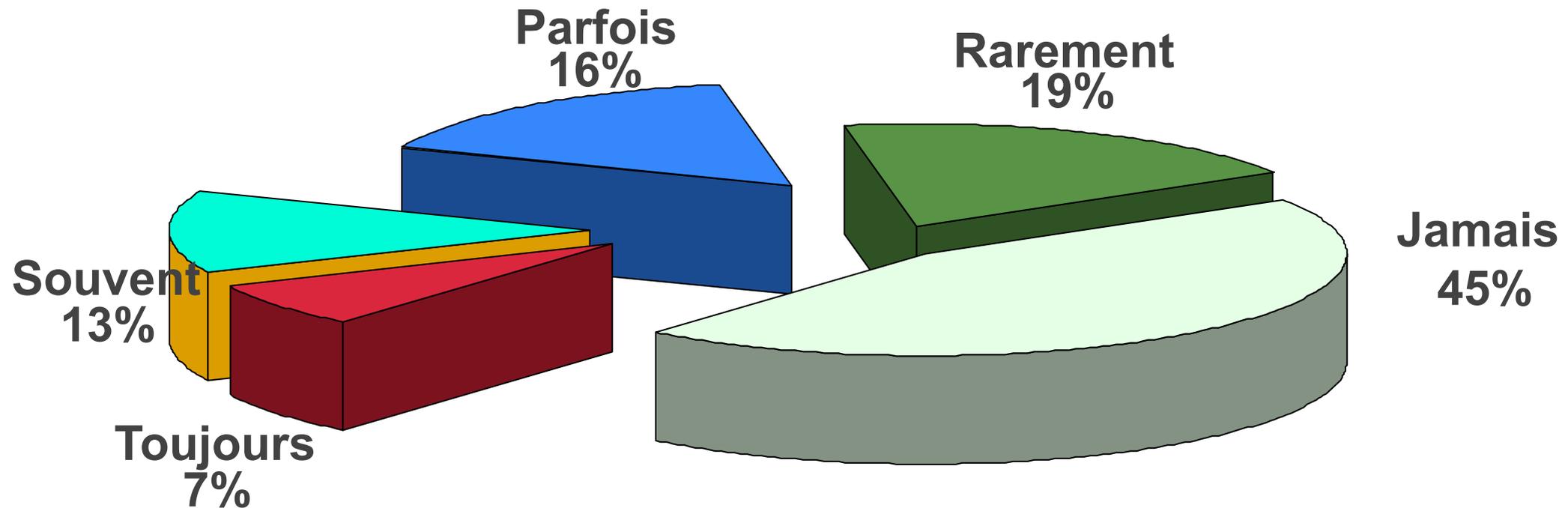


Racines de l'agilité 1



Dérives usuelles (1)

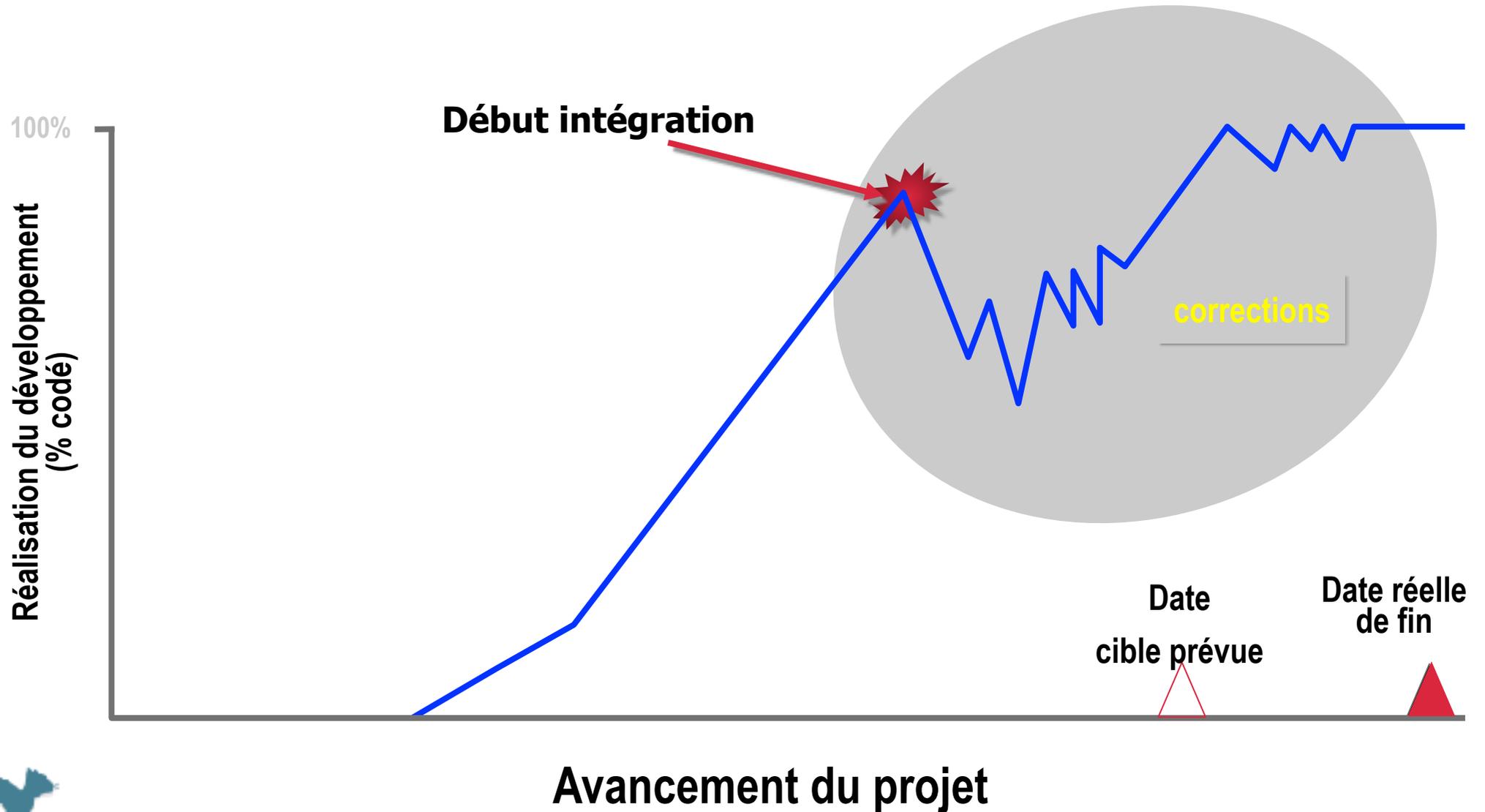
Fonctionnalités utilisées dans un système



Standish Group Study Reported at XP2002 by Jim Johnson,
Chairman



Dérives usuelles (2)



D'après Walker Royce

Manifeste Agile

Personnes et interactions

>

Processus et outils

Logiciel qui fonctionne

>

Documentation

Collaboration avec le client

>

Négociation à partir d'un contrat

Adaptation au changement

>

Suivi d'un plan



Source : www.agilemanifesto.org

Principes du manifeste

- ▶ Satisfaire le client en livrant tôt et régulièrement des logiciels utiles
- ▶ Le changement est accepté
- ▶ Livrer fréquemment une application fonctionnelle
- ▶ Collaborer quotidiennement
- ▶ Un logiciel fonctionnel est la meilleure unité de mesure de la progression du projet
- ▶ Rythme de développement soutenable
- ▶ Excellence technique et qualité
- ▶ La simplicité est essentielle
- ▶ Bâtir le projet autour de personnes motivées
- ▶ Conversation en face à face
- ▶ Equipes qui s'auto-organisent
- ▶ À intervalle régulier, l'équipe réfléchit aux moyens de devenir plus efficace



Sur le terrain

Manifeste agile

Valeurs

Principes

Génie
logiciel

Pratiques



Notion de succès

Optimise la valeur

Apporte de la valeur

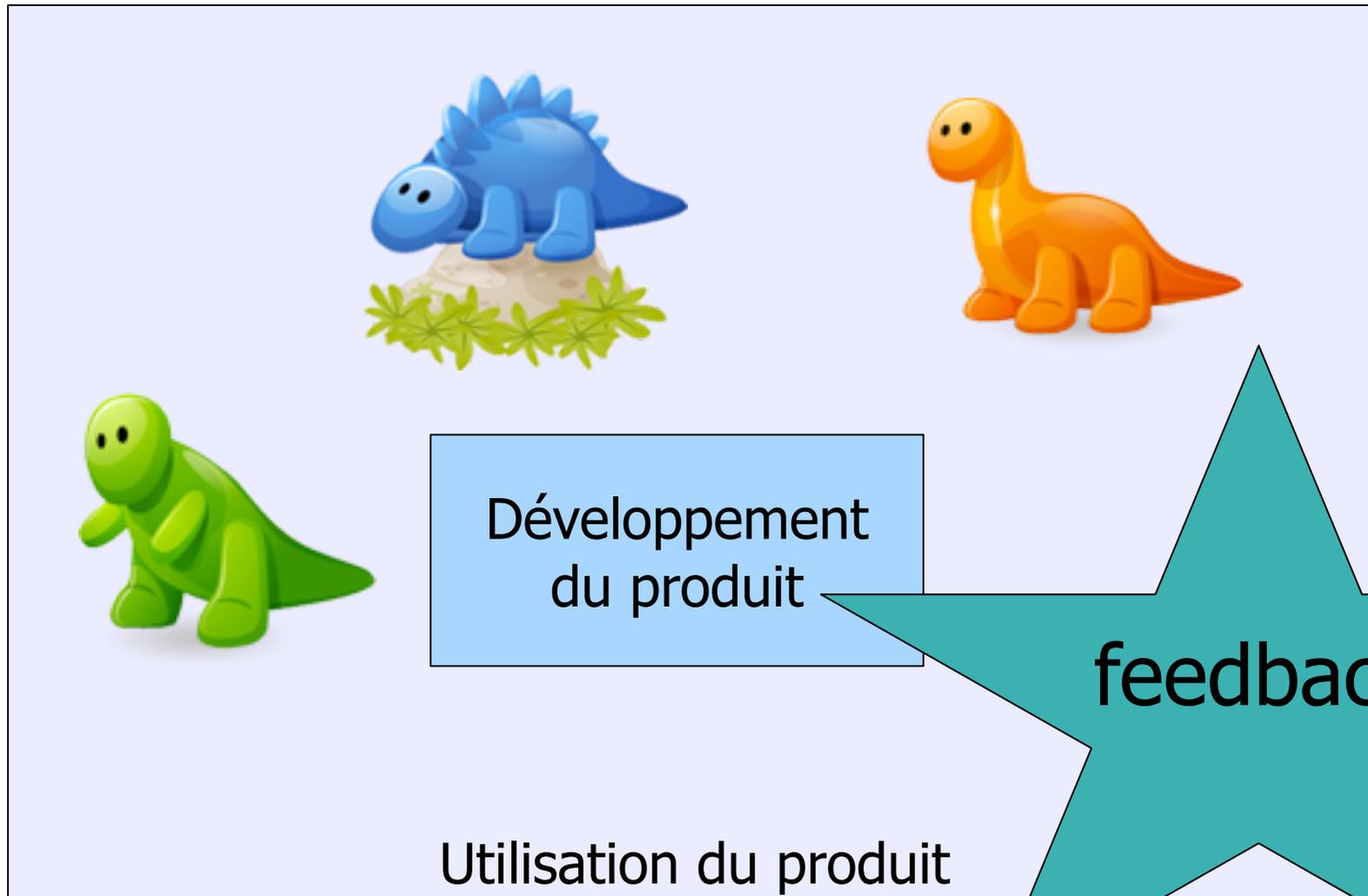
Répond aux besoins utilisateurs

Logiciel de qualité, maintenable

Logiciel qui fonctionne



Cadre élargi



Manifeste

- Valeurs
- Principes
- Logiciel

Mouvement

- Culture
- Croyances
- Pas seulement pour du logiciel

Méthode

- Scrum
- XP
- Kanban
- ...



Survól de Scrum

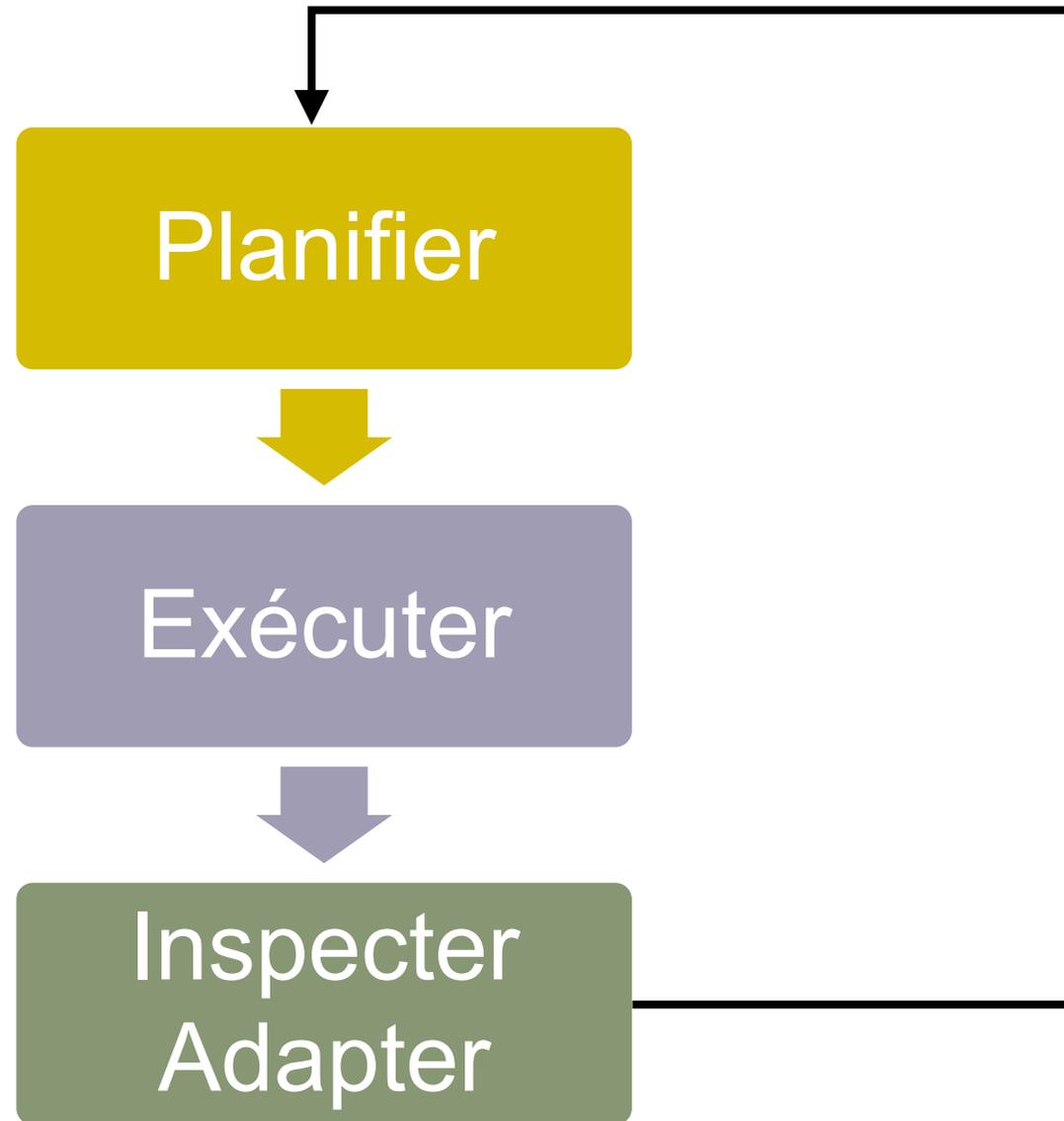


Survól de Scrum

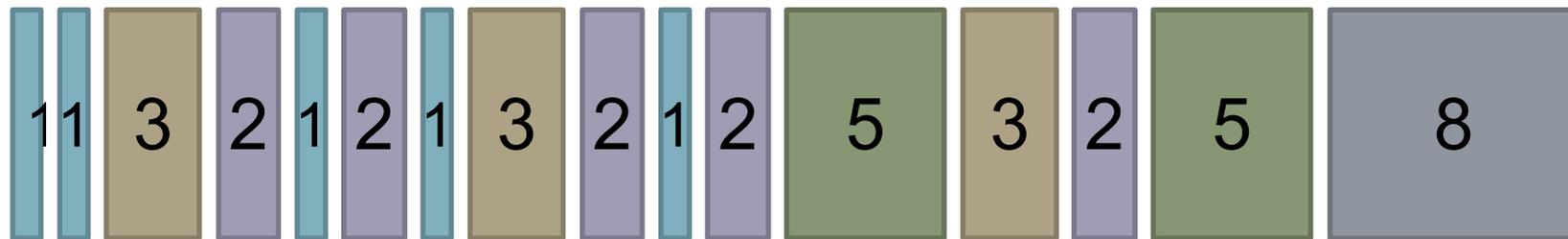




Théorie empirique de Scrum



Backlog



Équipe

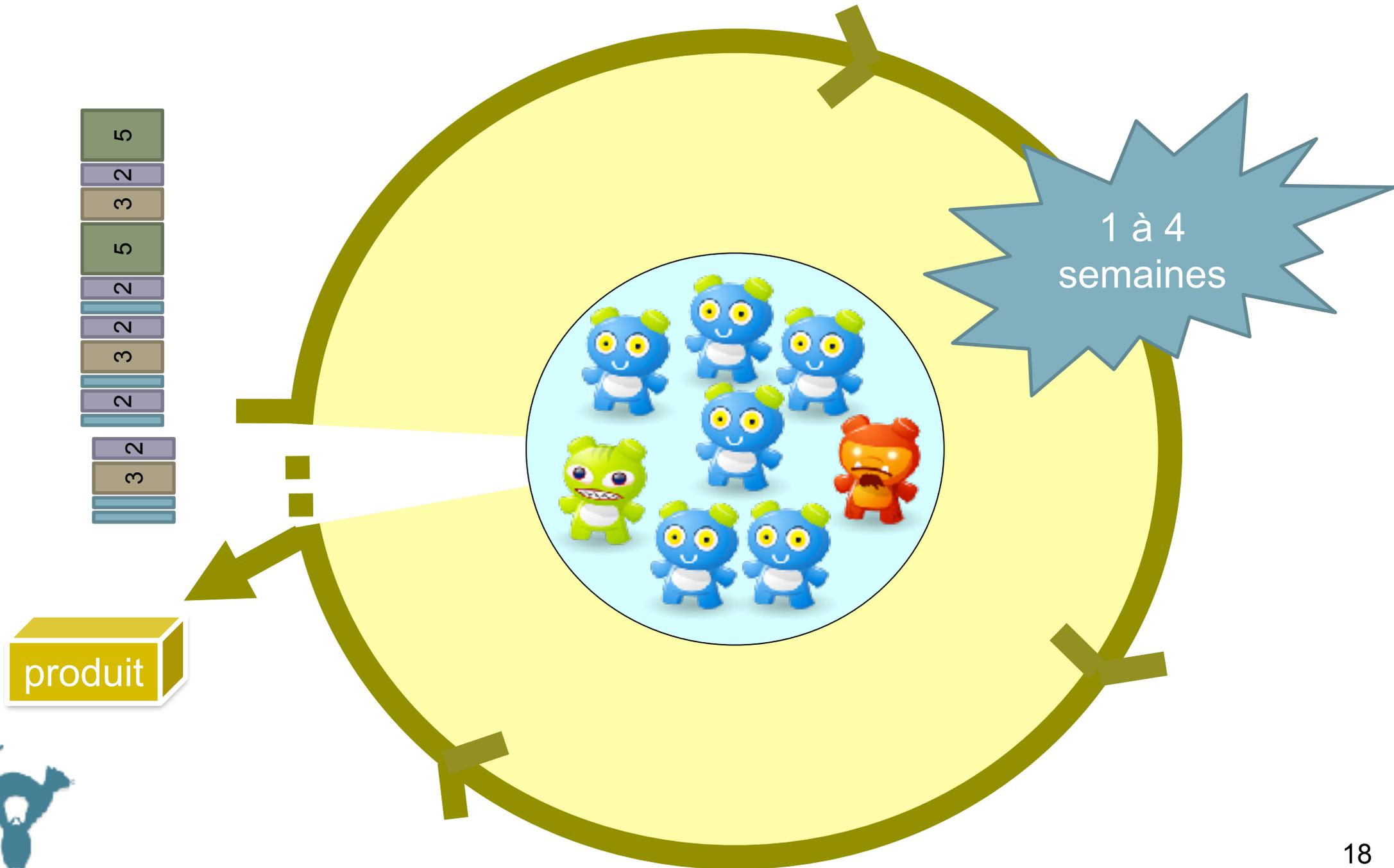
Product Owner



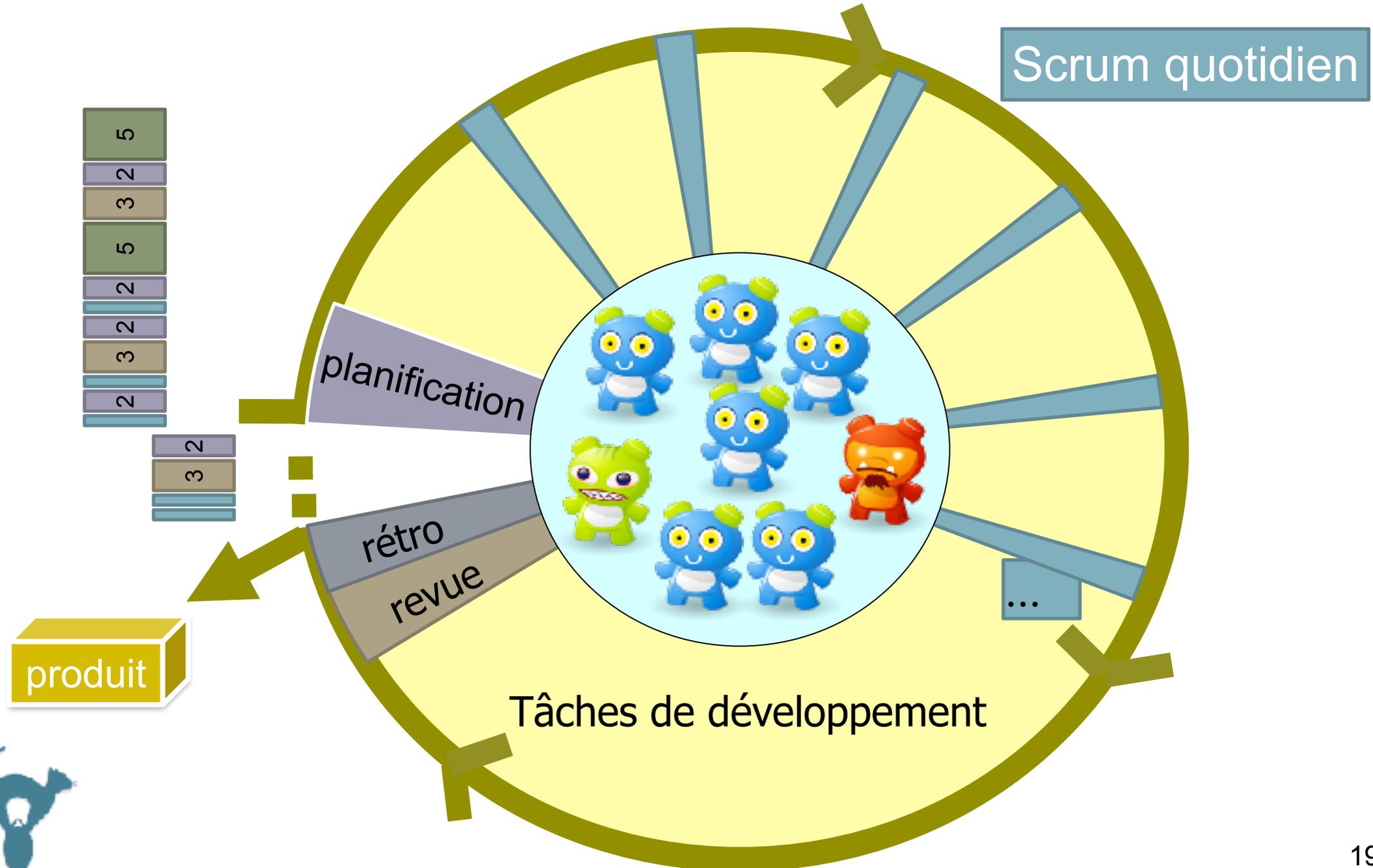
ScrumMaster



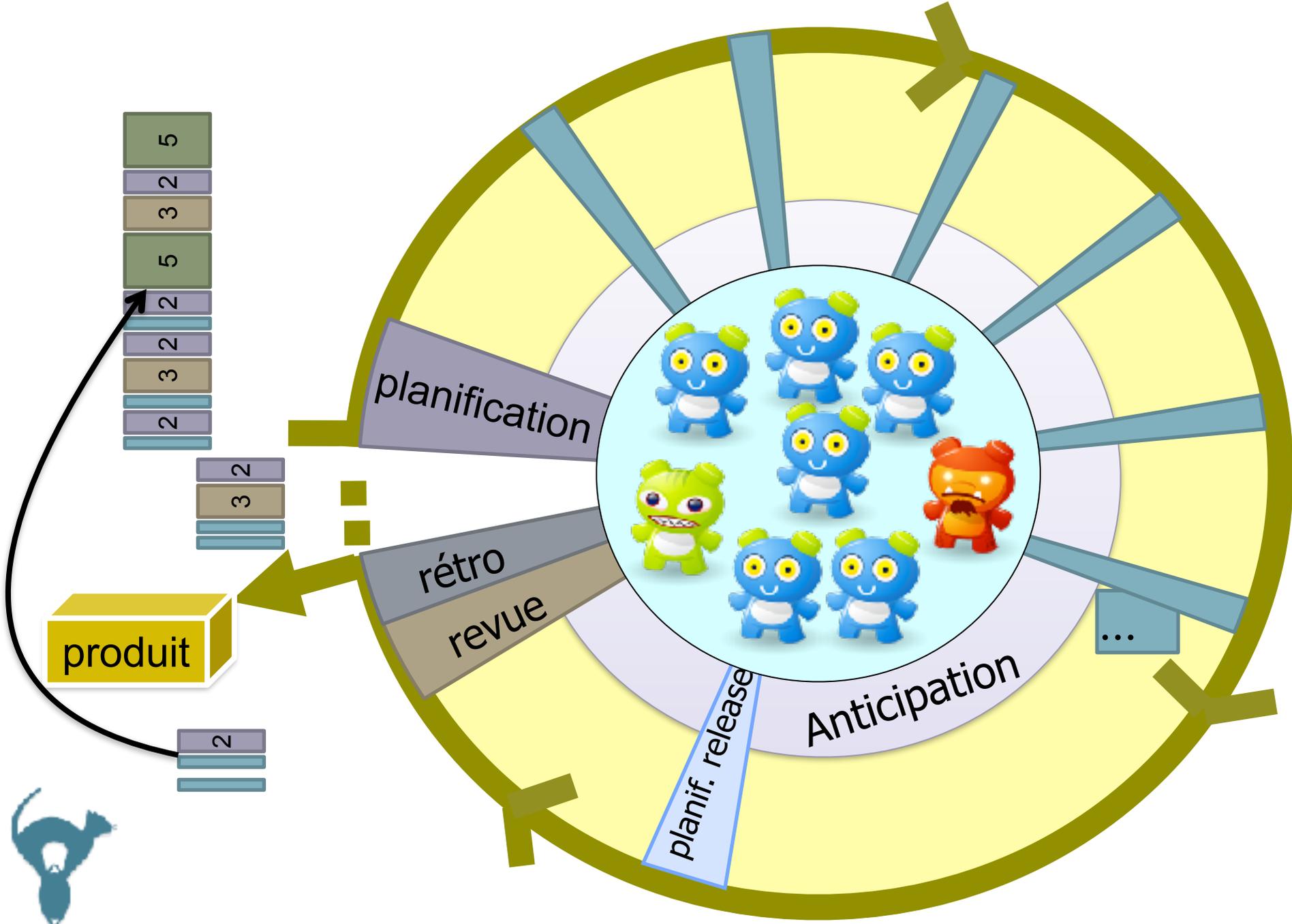
Sprint



Cérémonial



Feedback

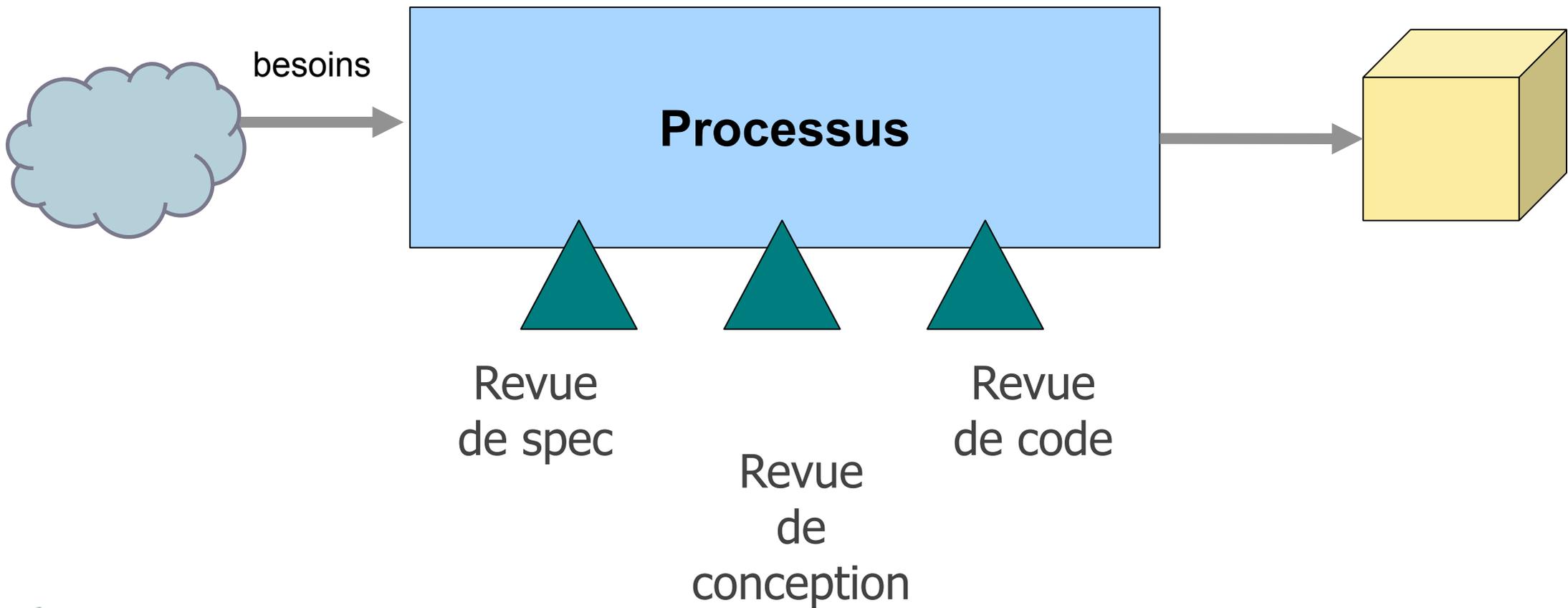


Des sprints pour une release

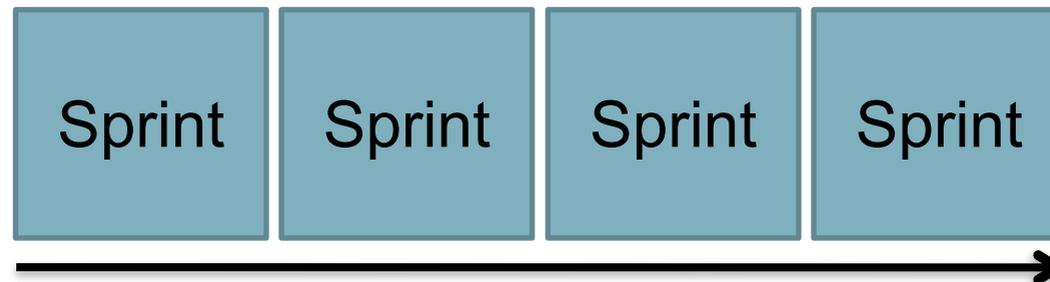
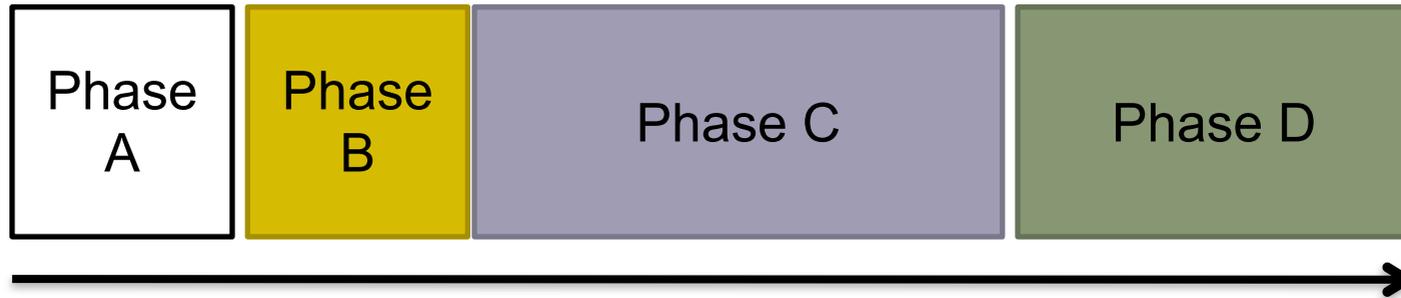
2



Effet tunnel

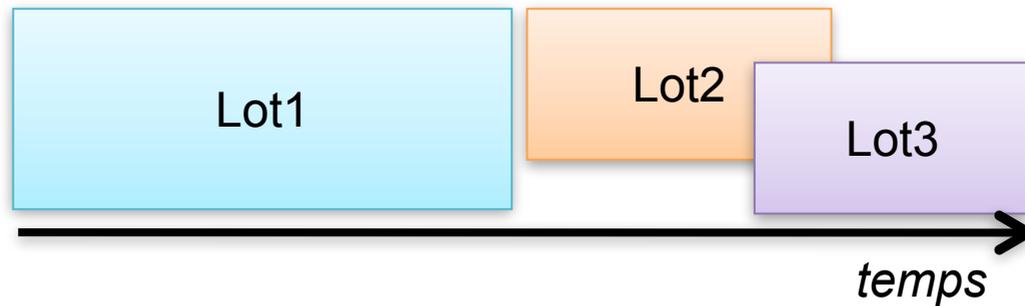


Phases

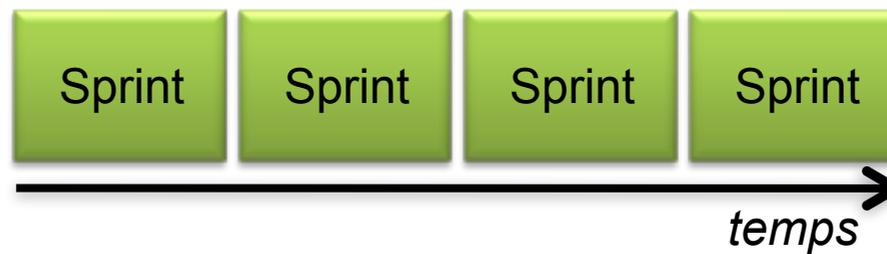


Incrémental n'est pas agile

Incrémental



Agile

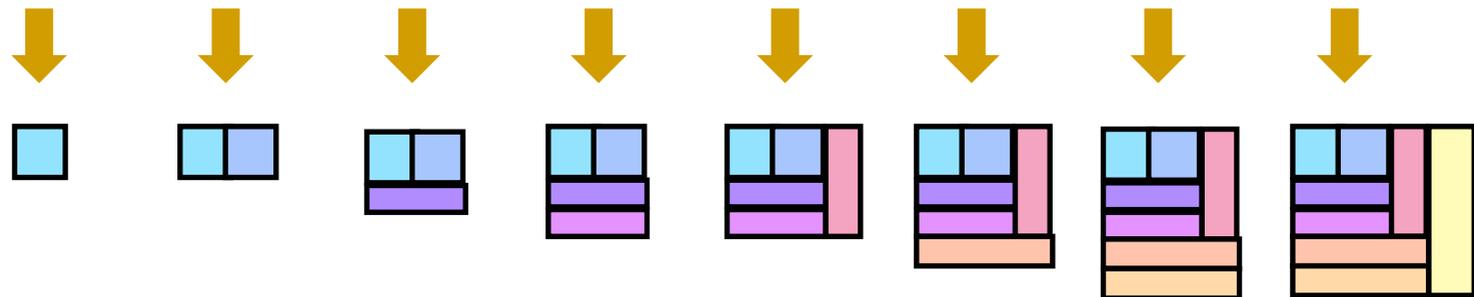


Résultat du sprint

sprints

Novembre

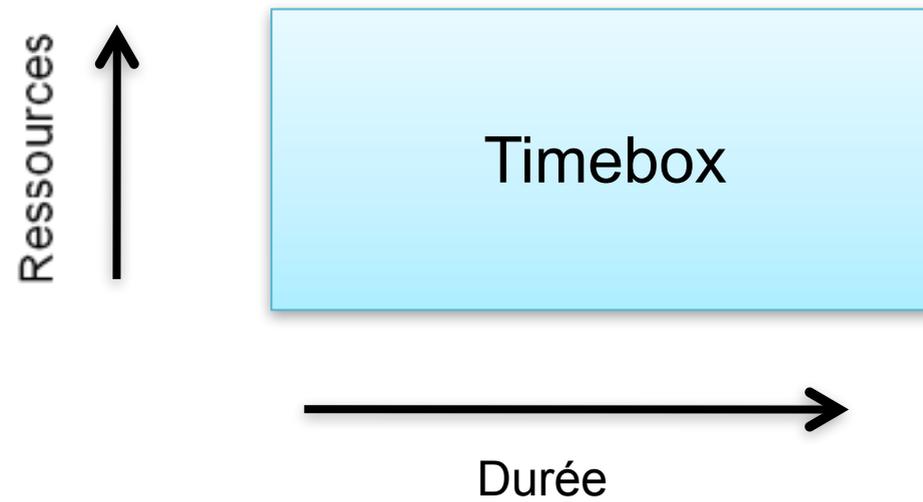
Avril



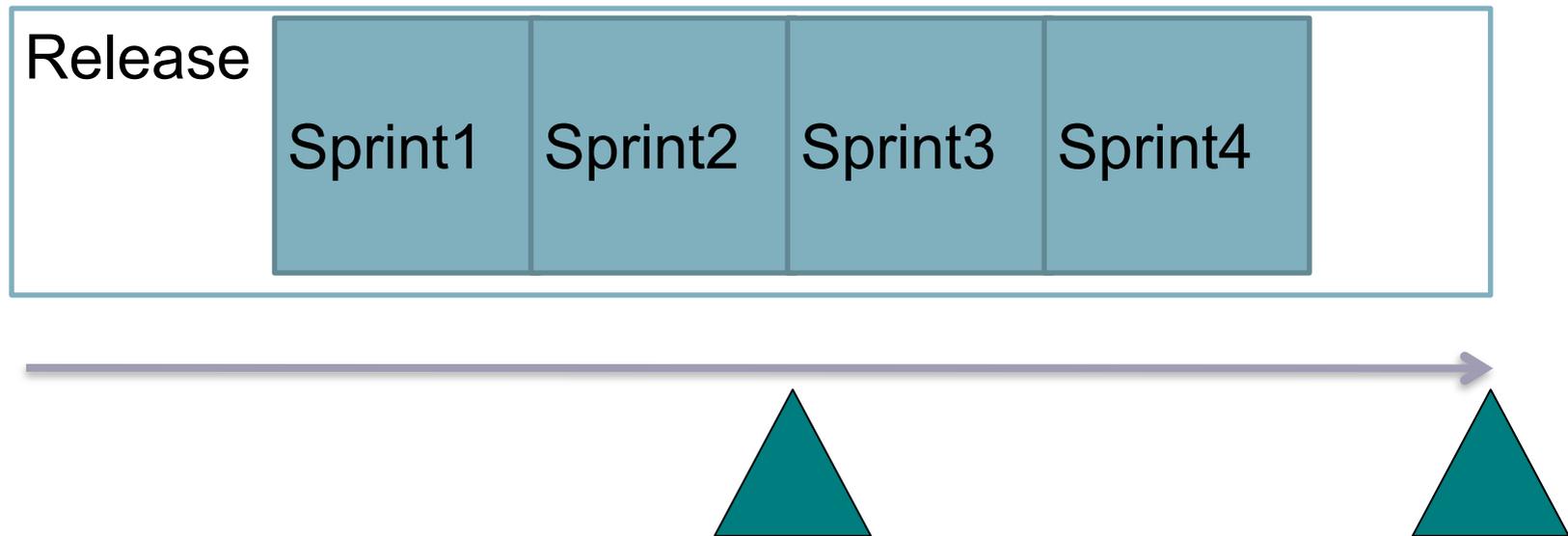
incrément



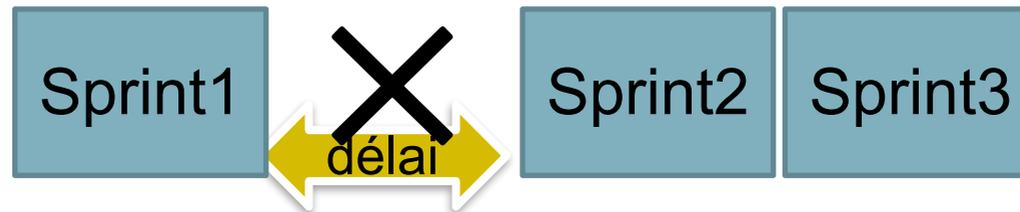
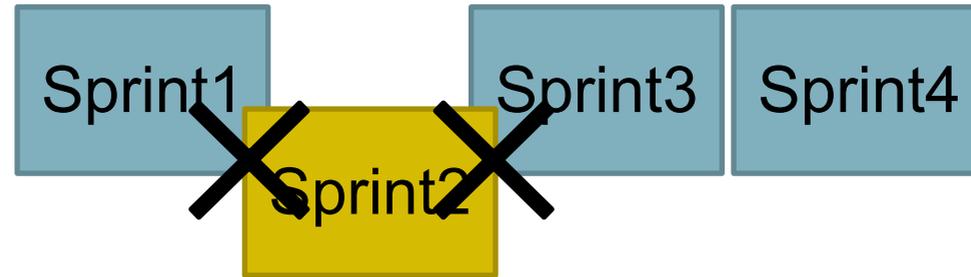
Boite de temps



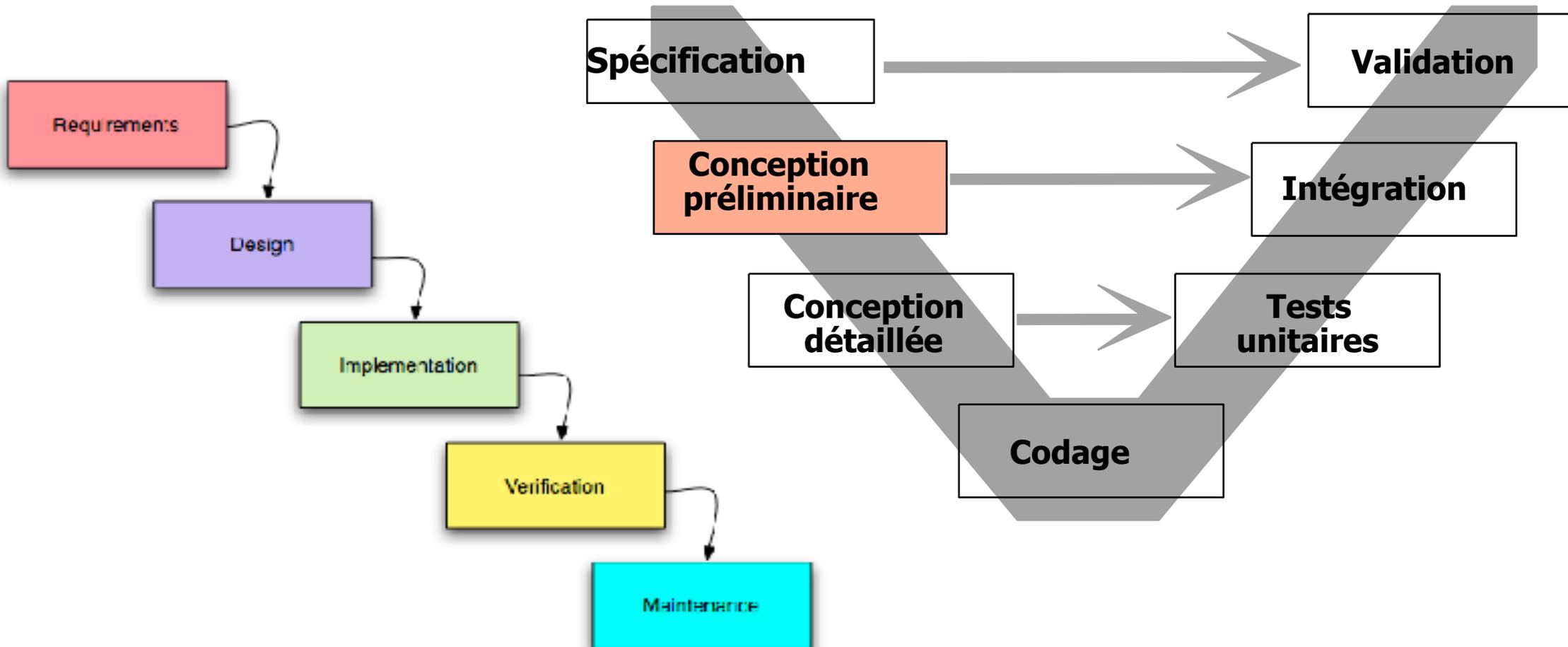
Une release



Des sprints qui s'enchainent



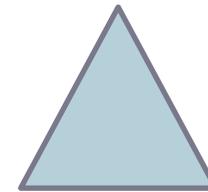
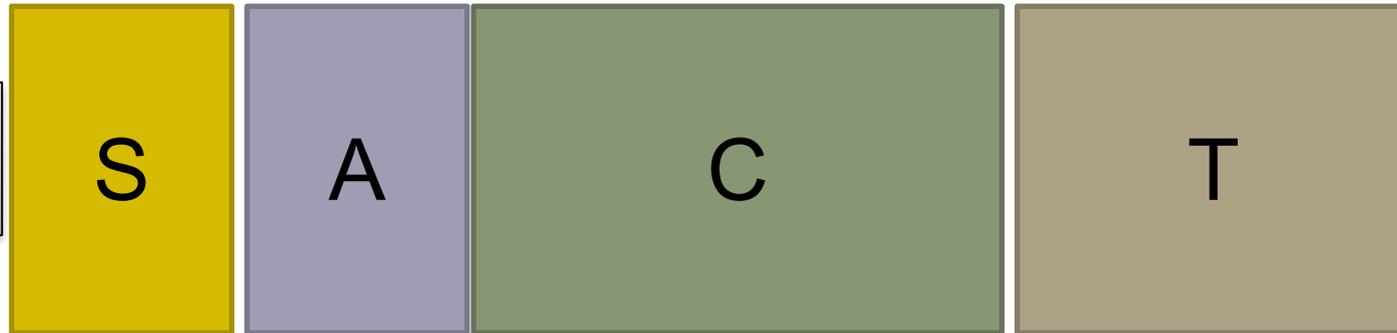
Cycles de vie ?



Activités en séquence

Une activité par phase ?

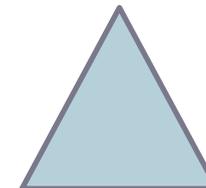
Théorie



Réalité

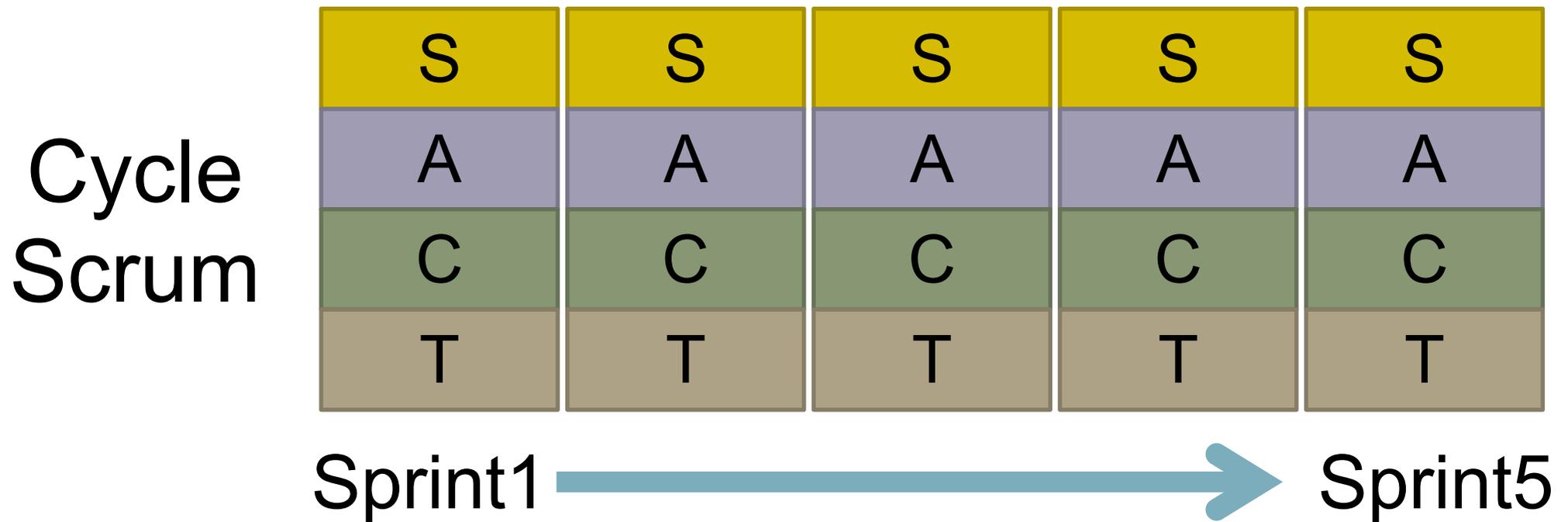


Retour sur les activités



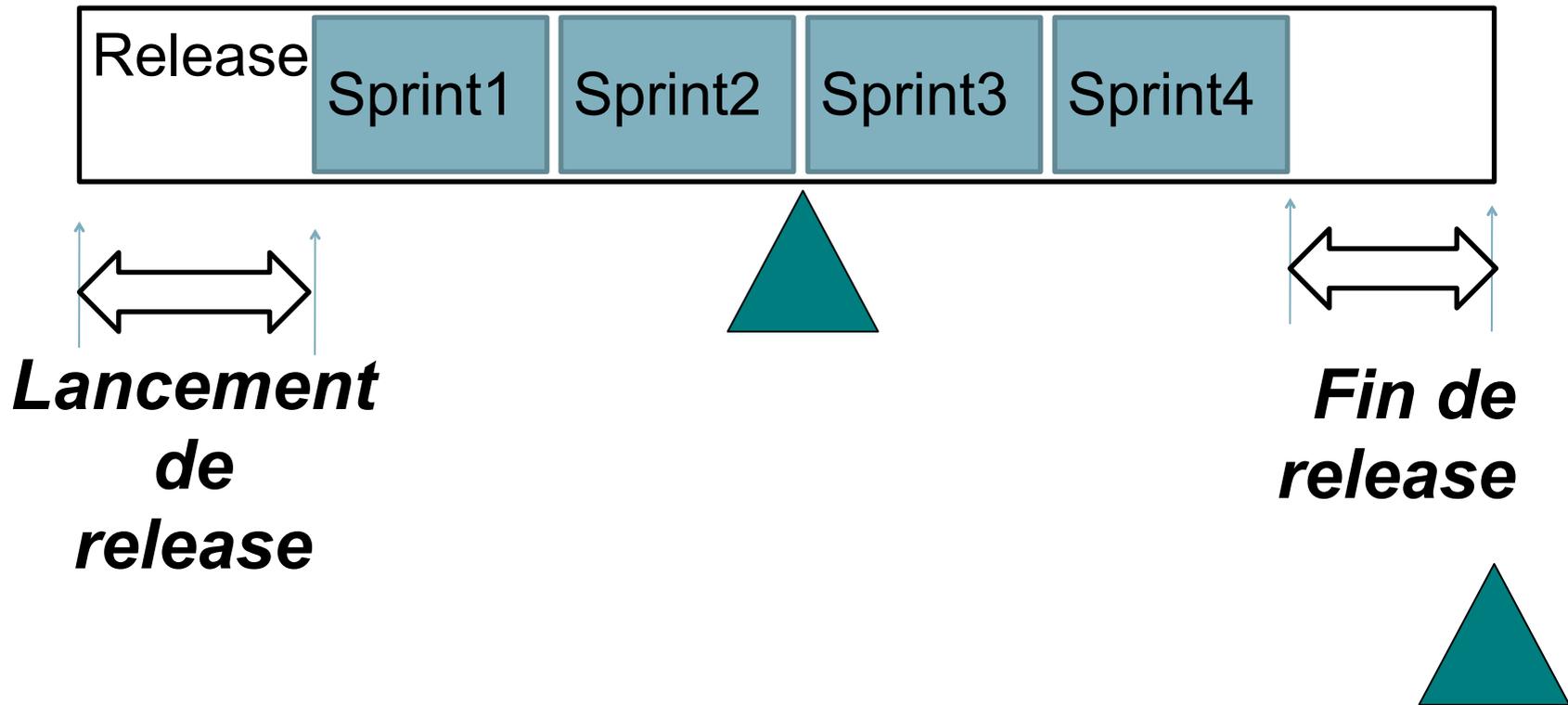
Activités en parallèle

Dans une journée, il est fréquent que le travail porte sur plusieurs activités

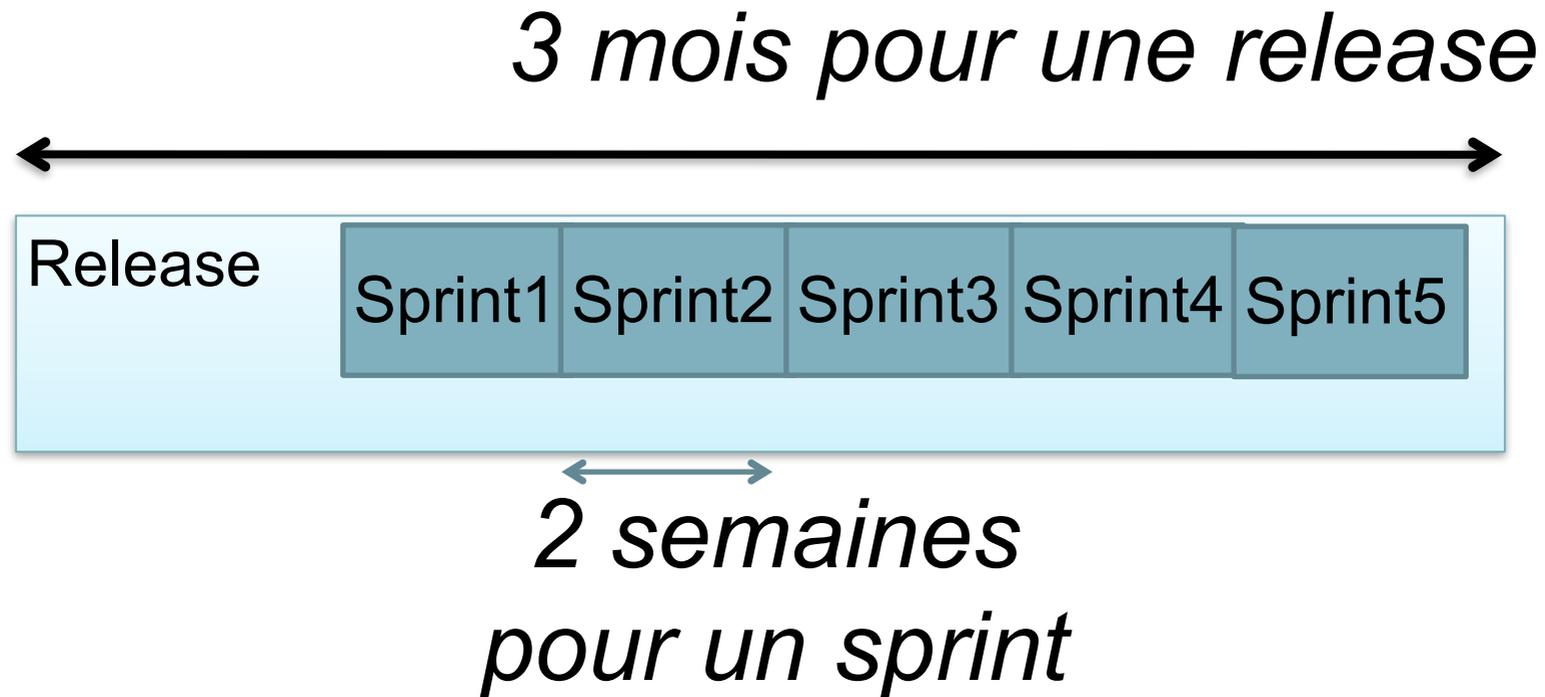


Les périodes d'une release

Les sprints



Un rythme typique



5 sprints dans une release



Le Product Owner

3



Implication du métier

Comme les utilisateurs ne savent pas comment résoudre leurs problèmes, nous devons arrêter de leur demander... il vaut mieux les impliquer

Mike Cohn



Le rôle

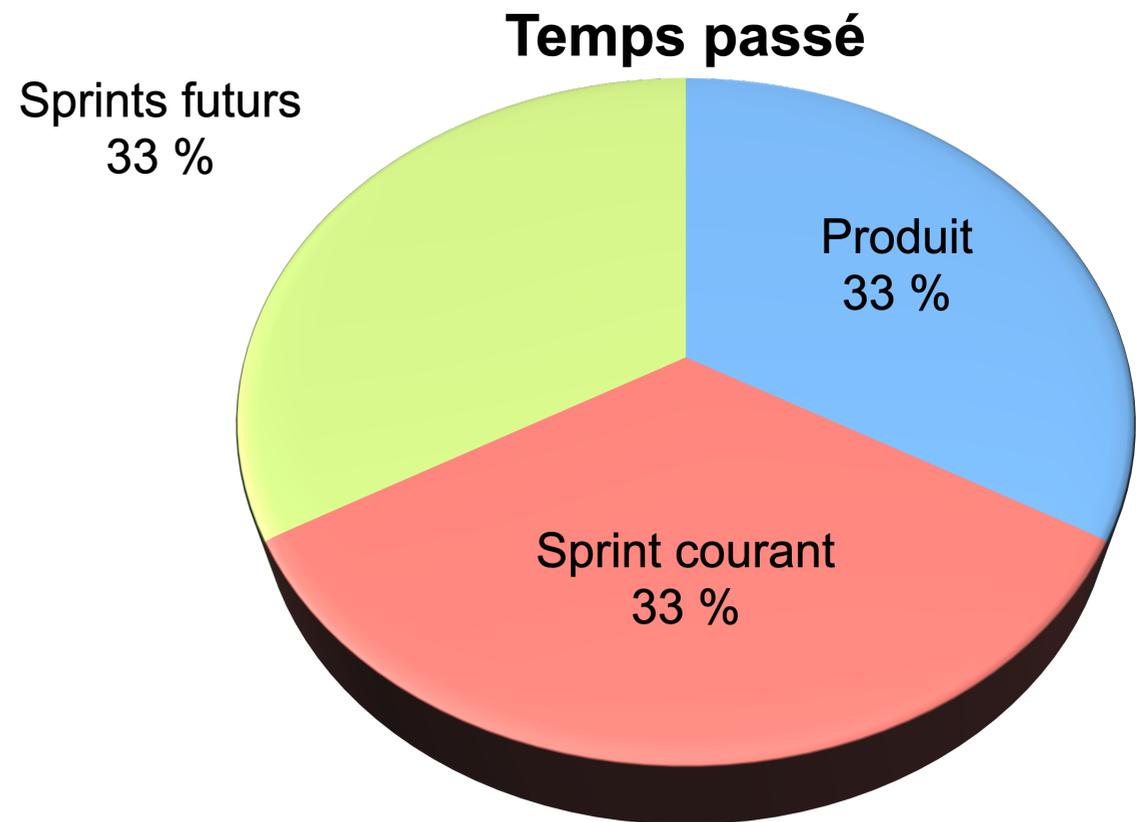
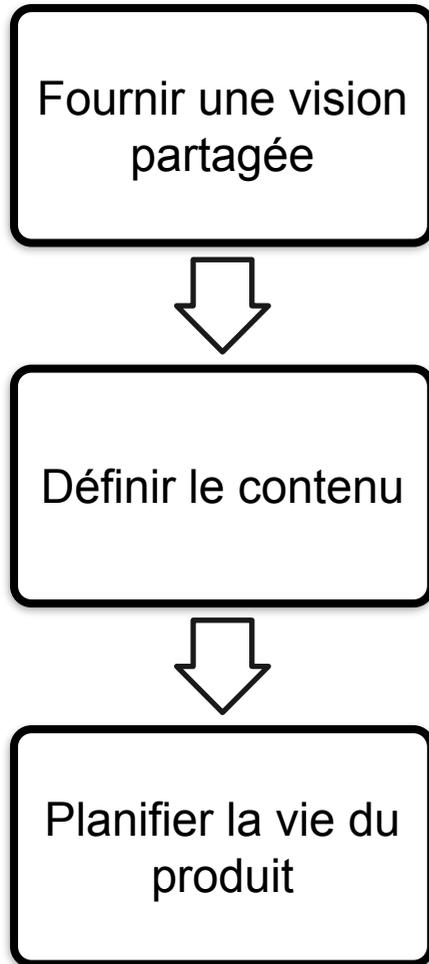
Dans une équipe Scrum qui développe un produit, le Product Owner est la personne qui représente les utilisateurs.



Il apporte sa vision à l'équipe et définit les priorités de façon à obtenir un produit ayant le maximum de valeur.



Répartition de son temps



Compétences souhaitées

Bonne connaissance du domaine métier

Maîtrise des techniques de définition de produit

Capacité à prendre des décisions rapidement

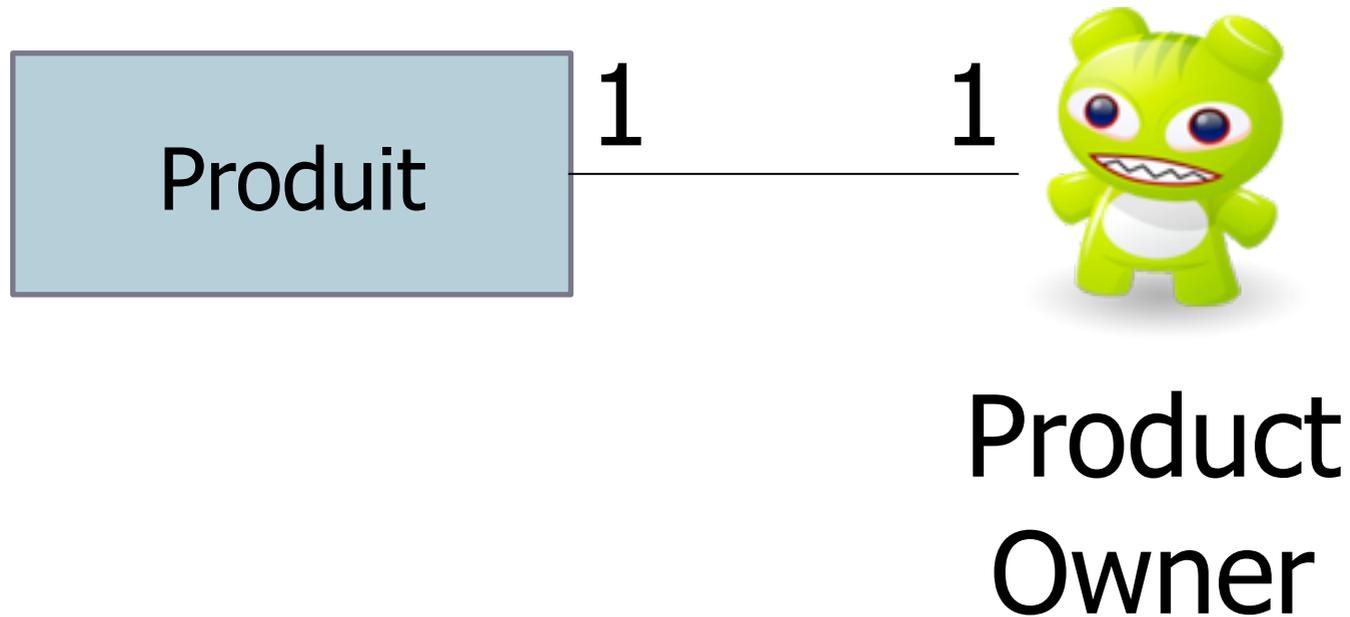
Capacité à détailler au bon moment

Esprit ouvert au changement

Aptitude à la négociation



Un « PO » par produit ?



Progresser dans le rôle

S'impliquer dans
les tests
d'acceptation

Collaborer avec
l'équipe



Cultiver le
backlog



Le ScrumMaster et l'équipe Scrum



4



Les rôles

L'équipe est investie avec le pouvoir et l'autorité pour faire ce qu'elle a à faire, elle s'organise par elle-même.



Le ScrumMaster n'est pas un responsable hiérarchique, c'est un facilitateur.



Equipe responsable



- ▶ Effets de la tendance au taylorisme
 - ▶ Des rôles trop spécialisés
 - ▶ Une gestion excessivement axée sur l'individu
 - ▶ Des mesures inadaptées

Le développement est un travail collectif



Les membres de l'équipe



- ▶ Un groupe de personnes regroupant toutes les compétences nécessaires pour faire le travail
 - ▶ Architecture, conception, développement, IHM, test, documentation, etc.
- ▶ Recommandations
 - ▶ Membres à plein temps sur le projet
 - ▶ Équipe stable



Extrait du Midol

...une distribution des joueurs les plaçant dans des situations de jeu où ils devraient pouvoir développer des compétences autres que celles de leur poste.

L'investissement de chaque joueur dans ces tâches qui ne relèvent pas du poste, se doit de répondre au contexte situationnel donc au sens attribué par toute l'équipe à cette situation.

On peut même dire quand tout est bien huilé et que chacun joue juste dans le rôle qui lui est momentanément imparti dans la situation existante que cette « synchronisation forcement collective » constitue alors un véritable système de jeu qualifié d'**agile**



Responsabilités du ScrumMaster

Fait appliquer les valeurs et les pratiques de Scrum

Elimine les obstacles qui empêchent l'équipe d'avancer

Motive l'équipe pour l'atteinte des objectifs

Protège l'équipe des interférences extérieures



Etat d'esprit du ScrumMaster

Capacité à percevoir les émotions dans l'équipe

Curiosité et envie d'apprendre

Inclination à penser que les gens font de leur mieux

Envie de changer les choses qui ne vont pas bien

Orientation vers le collectif

Prise de risques



Progresser dans le rôle

Parfaire sa connaissance de Scrum

Favoriser l'auto-organisation de l'équipe



Analyser les causes des problèmes

Maîtriser le reporting

Pratiquer l'art du possible



Les intervenants

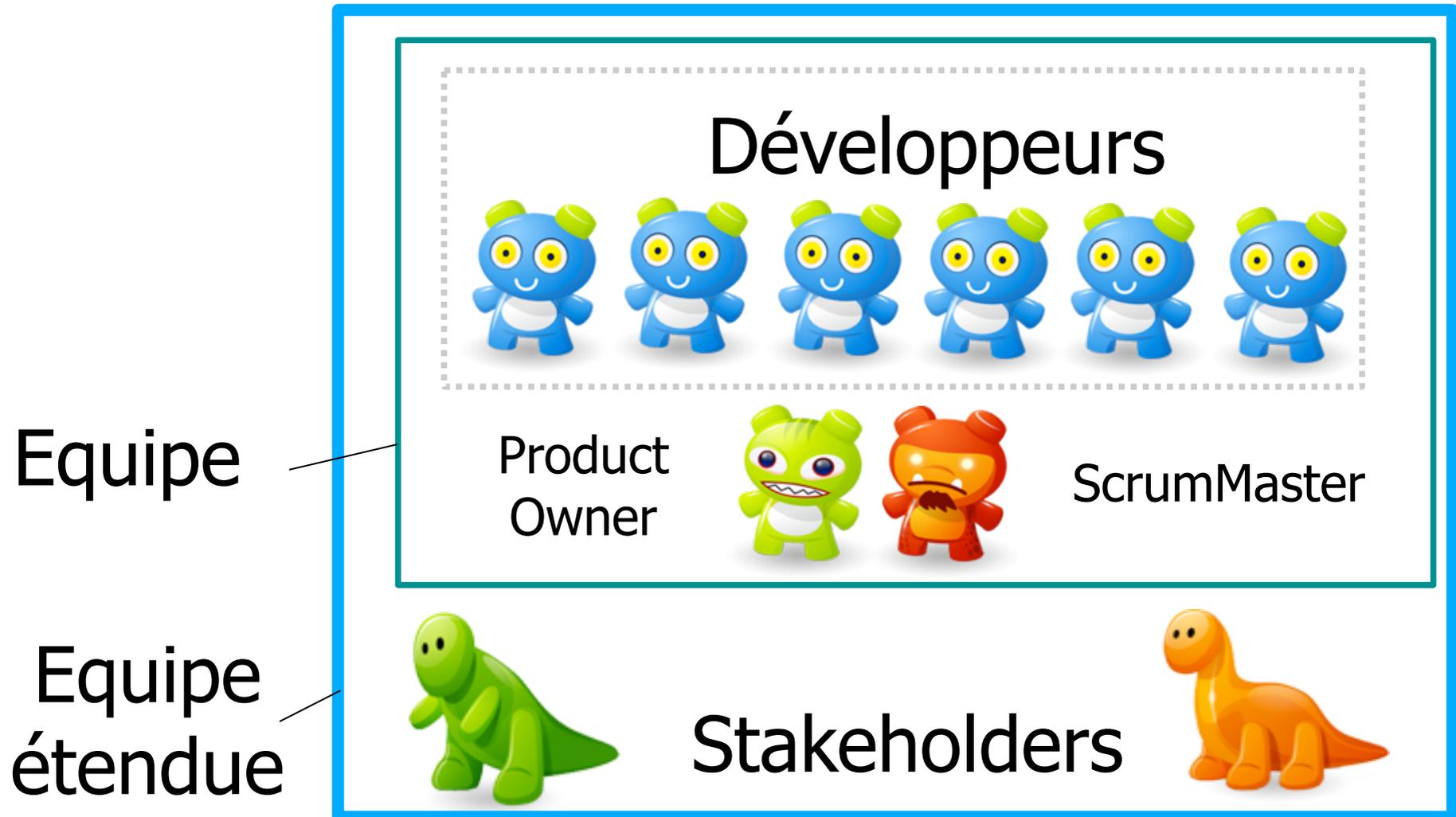
- ▶ Toutes les personnes impliquées dans le projet et son résultat
 - ▶ Utilisateurs
 - ▶ Clients
 - ▶ Personnes chargées de la maintenance
 - ▶ Infrastructure et opérations
 - ▶ Marketing, analystes métier
 - ▶ Managers



Stakeholders



Equipe complète



Equipe idéale

Equipe développement

- Entre 3 et 8 développeurs, avec au moins un qui a de bonnes compétences en architecture
- Au moins un testeur (**analyste**) pour 3 développeurs
- D'autres spécialistes si nécessaire : IHM, BD

Equipe projet

- Un et un seul Product Owner
- Un et un seul ScrumMaster

Equipe complète

- Plusieurs utilisateurs
- Le management
- Les services transverses

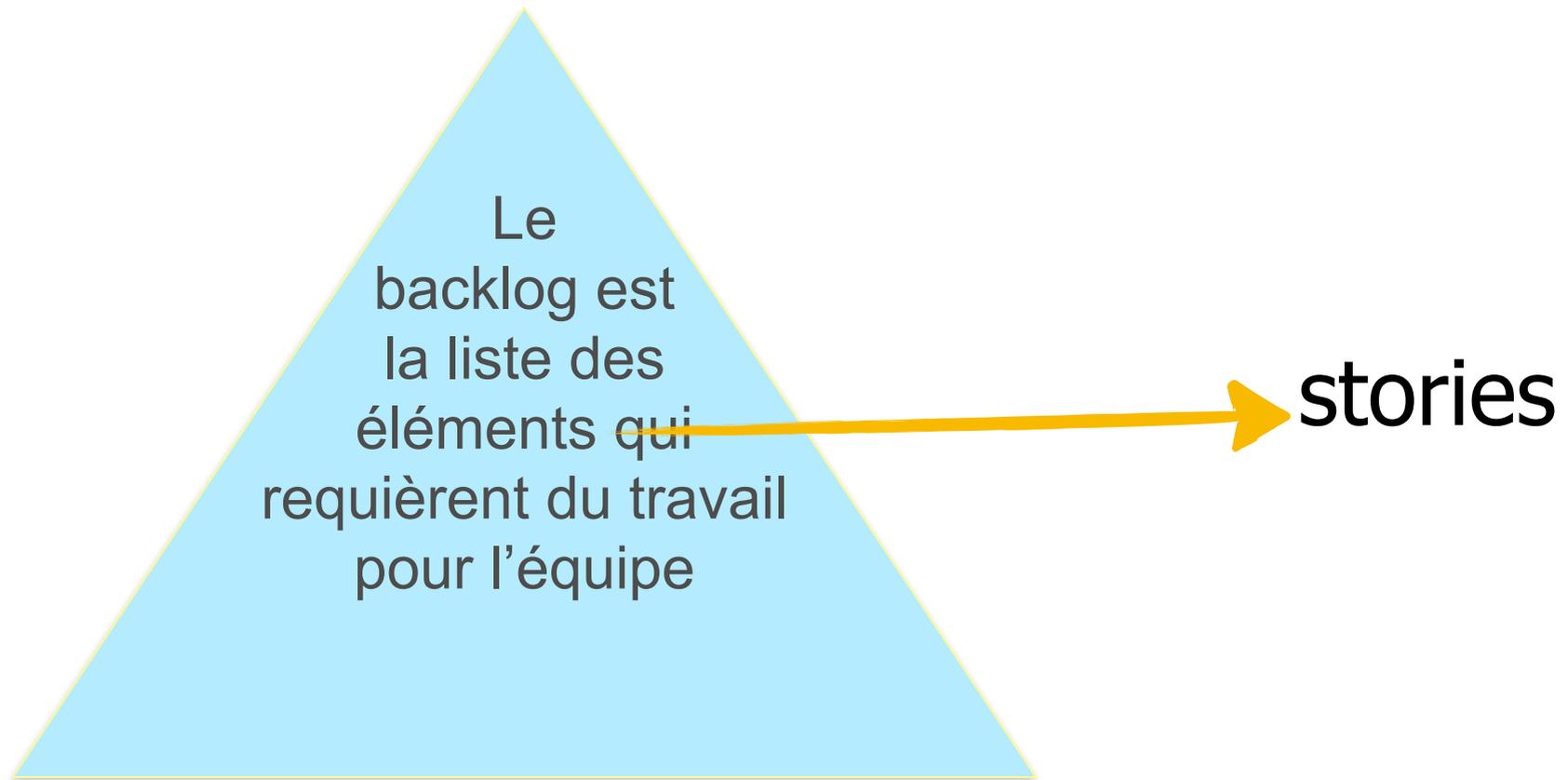


Le backlog de produit

5



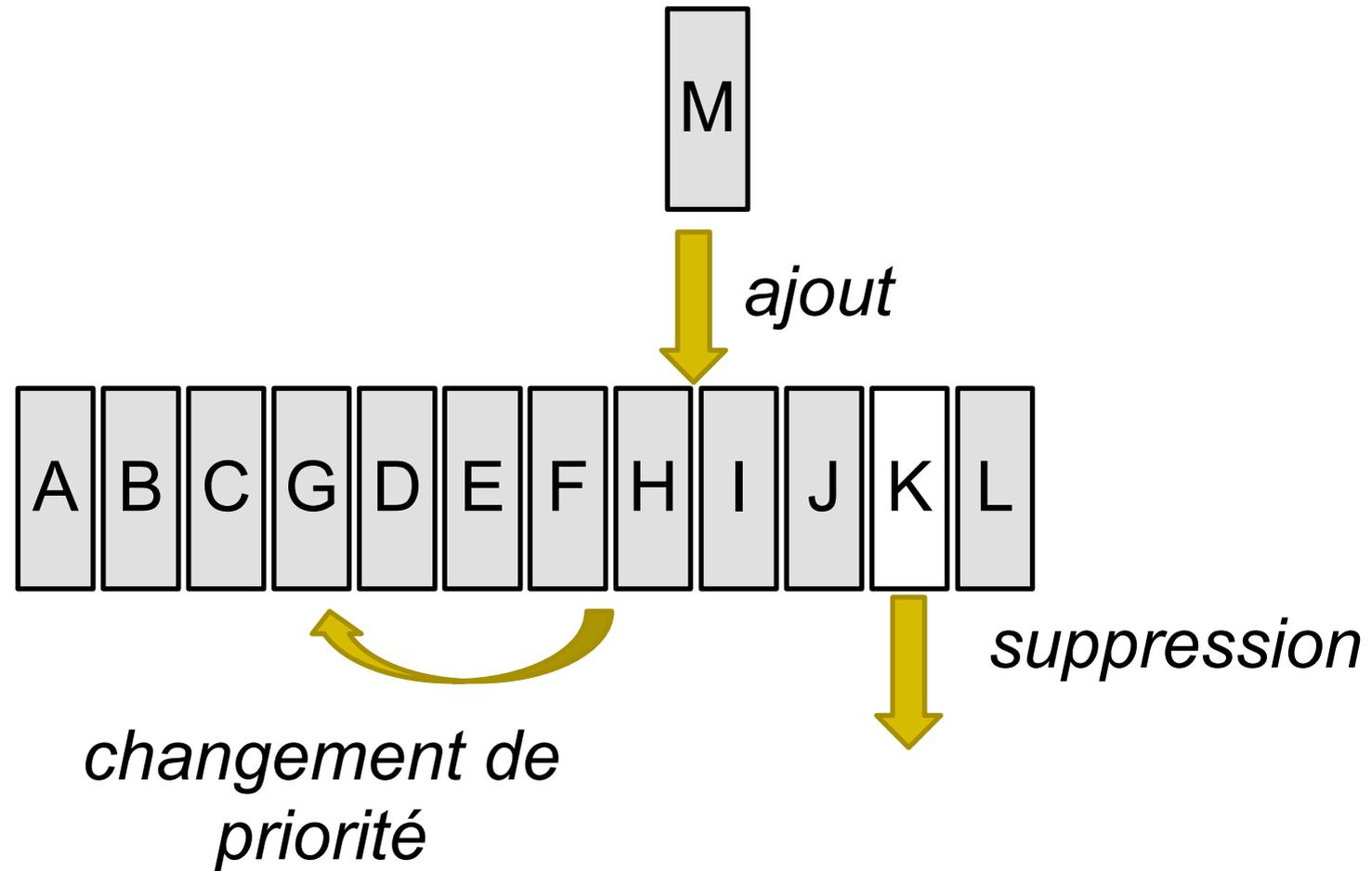
Définition



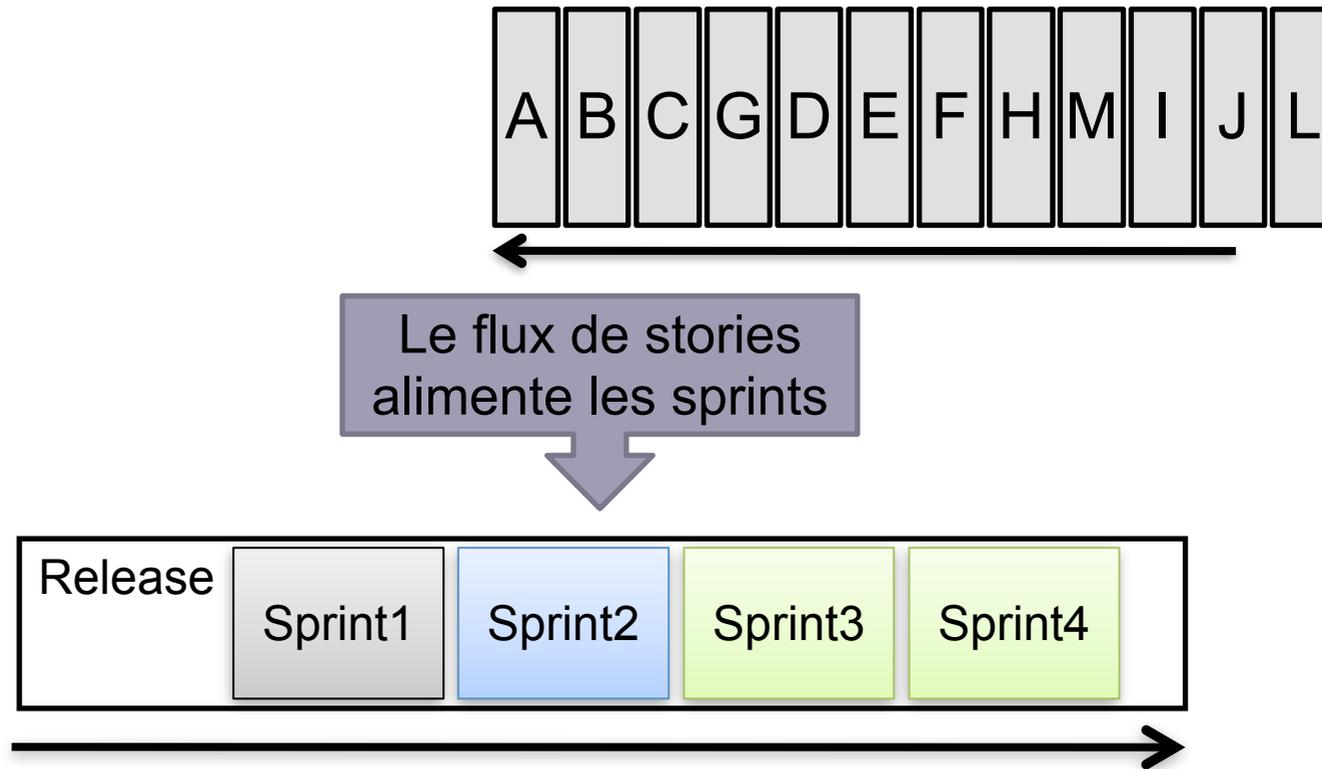
Une liste unique



Une liste vivante



Priorités



Critères pour la priorité

La valeur ajoutée (l'utilité)

Le coût de développement (l'effort)

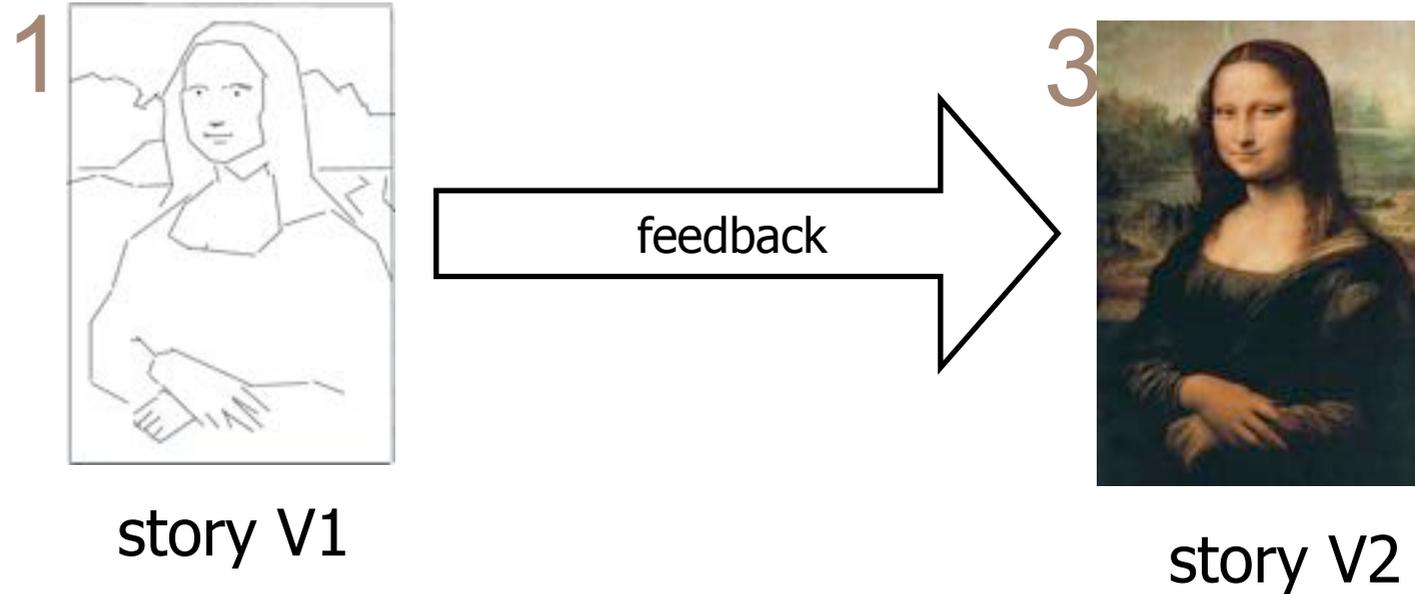
La connaissance apportée

La quantité de risques éliminée

Les dépendances



Apport de connaissances



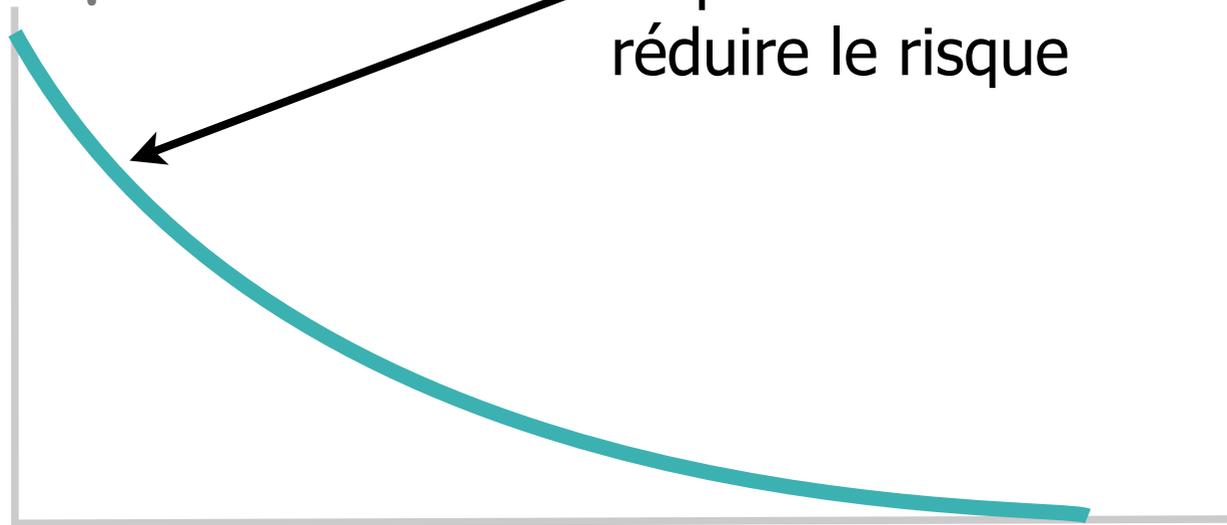
D'après Jeff Patton



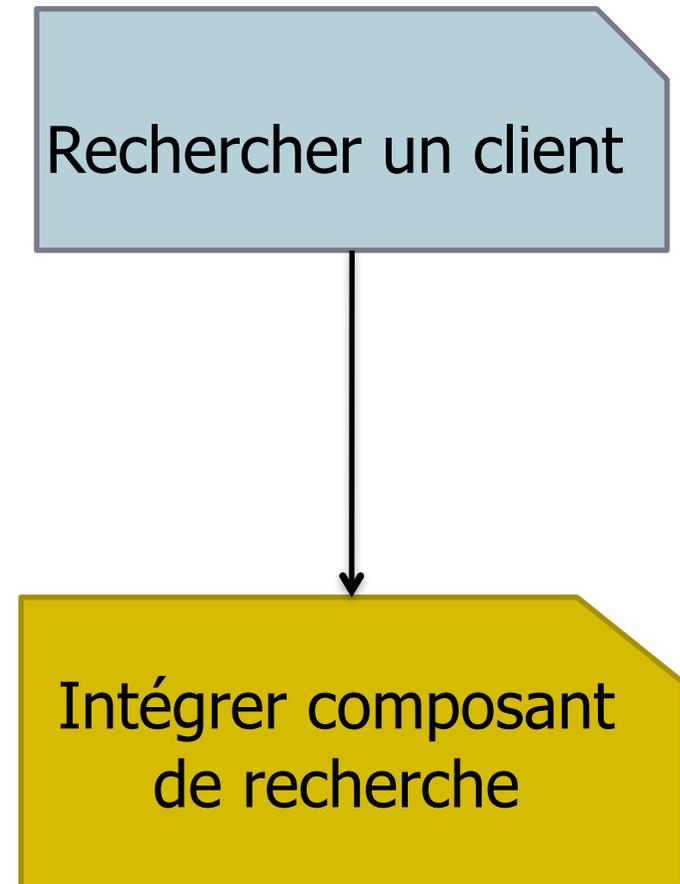
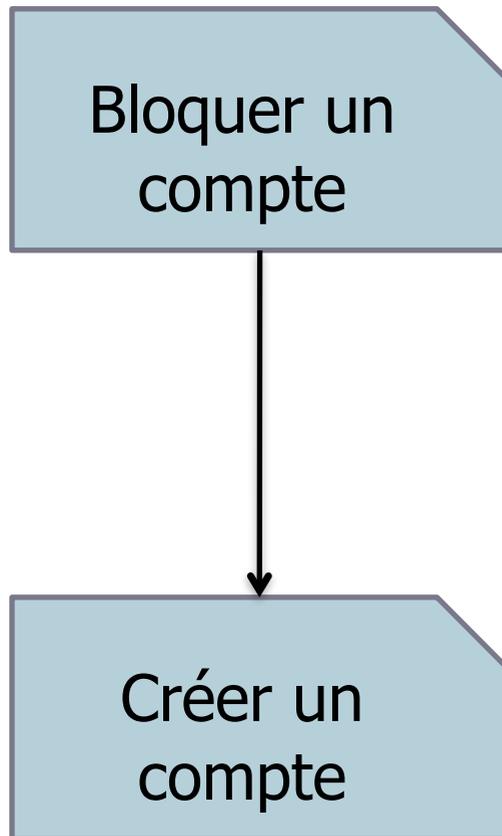
Risque

Exposition
au risque

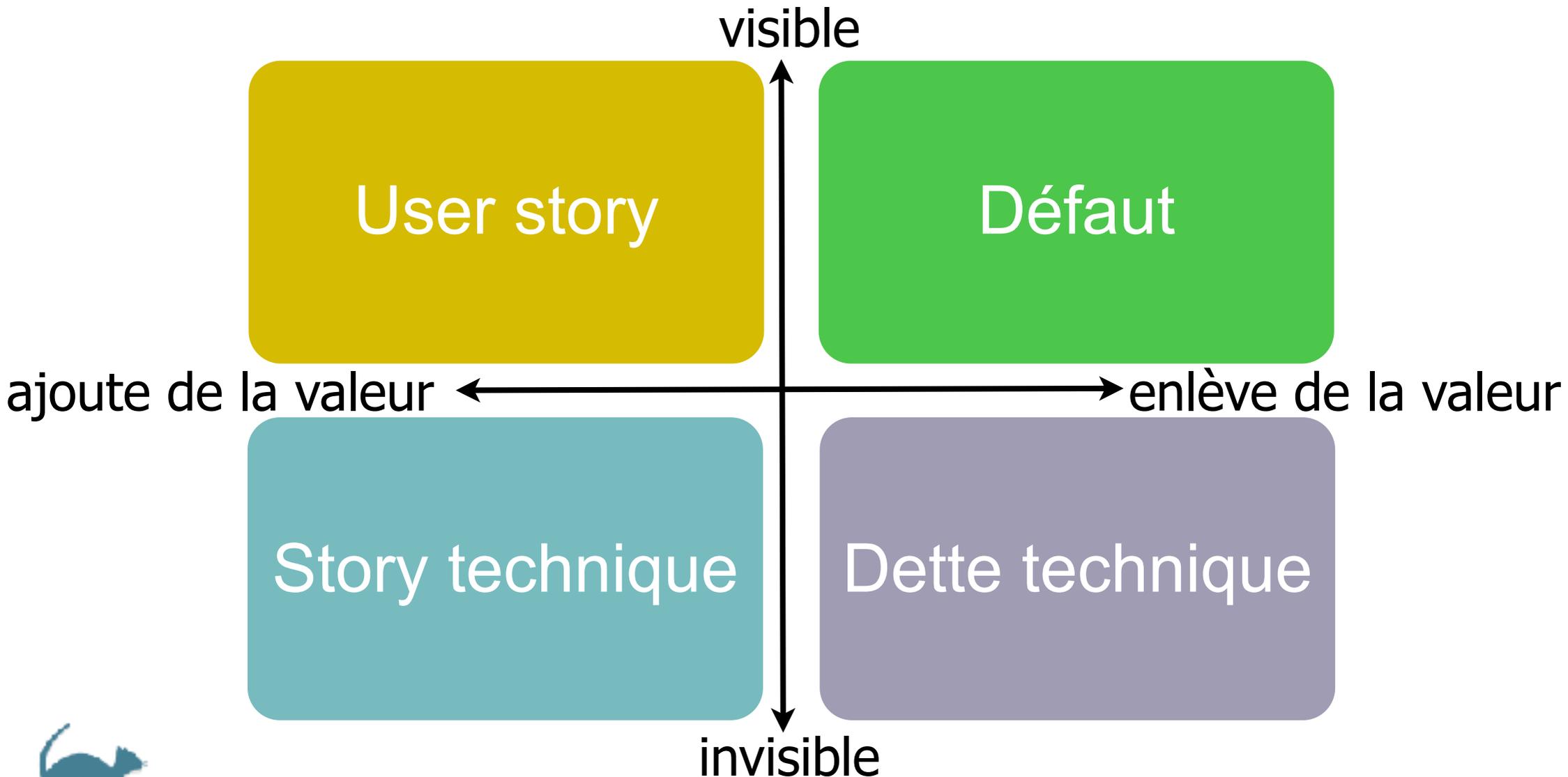
story qui
permet de
réduire le risque



Dépendances



Type des éléments du backlog



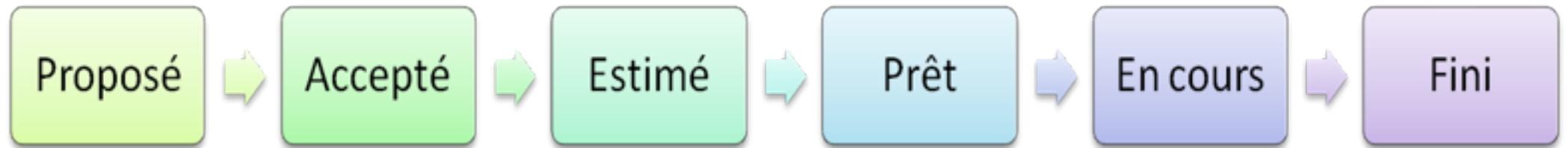
Attributs d'une story



Il existe un plan-type



Workflow d'une story



Bien gérer le backlog

A essayer

- Le construire collectivement
- Le faire vivre et le maintenir prêt
- Décomposer progressivement
- Gérer les priorités par un ordre de présentation
- Nettoyer en profondeur à chaque fin de release

A éviter

- Dépasser 50 éléments dans le backlog
- Utiliser un bugtracker en plus du backlog
- Attendre d'avoir une spécification détaillée avant de commencer le backlog
- Avoir plusieurs backlogs pour une seule équipe



En résumé

348 Task with multipl... As a Team Member I want to have a possibility to s...	347 Provide mechanism... From time to time a PM (or other user) may change ...	346 Parallel Sprints ... As Product Owner I want to be able to have paralle...	345 Filtrer le bac à ... En tant que utilisateur d'iceScrum, je peux affich...	344 Sort tables As a user I want to sort the tables by different c...	343 Change column wid... As a user I want to change the width of any column...	342 Exporting PNGs do... Try to export a (burndown)chart i.e. select "save ...	341 Update Release mi... Planned Release from 7.Jan.2011 to 31.Mar.2011; Wa...
340 Profiter des plan... En tant que membre de l'équipe SCRUM, je peux déco...	339 Backlog vers tâch... De la même façon qu'on peut recopier rapidement un...	335 Insérer des référ... Il serait utile de pouvoir insérer des références ...	334 Les stories s'aff... Dans la vue par sprint, il serait bien que les sto...	332 Paramétrage de la... En tant que Scrum Master, je peux paramétrer la li...	210 Update indicators Some indicators are only updated at the sprint act...	207 Sprint & Release ... Same as for sandbox or backlog in table format to ...	206 Webservice API Offer webservices for interact from/within other de...
205 LDAP authentificati... As User, I can register/login using a LDAP account...	199 Import tasks from... Import tasks from Excel with their estimations. A...	180 Pb mise à jour qu... Bonjour, suite à une réorganisation de notre proje...	174 Blocked and Ready... It would be nice to have extra columns in the Spri...	169 Indiquer le nombr... A la place d'avoir "o" sur le Reste à faire d'un ...	160 Sort table by ind... In each table I need to sort the table by any of i...	159 Change height of ... In all tables I need to change the height of each ...	158 Change width of c... In all tables I need to change the width of each c...
157 Available resourc... I need to specify the available resources (working...	156 Total hours of al... During sprint planning I need to know the total ho...	153 Disciplines for t... In a production environment, you often have differ...	152 Impediments manag... Provision to add/modify/update impediments is reqd...	137 Parallel releases As a product owner, I can create releases that run...	135 Parent / Child St... The ability to be able to associate a parent or a ...	133 Filtre pour Produ... Quand je regarde le backlog, parfois j'aimerais vo...	131 Compte du nombre ... Compter le nombre de tâches urgentes (récurrentes)...
130 Proposer des TAF ... POur toutes les tâches urgentes à faire et en cour...	120 Collecte des ress... Demande au moment de la clôture pour chaque membre...	119 Collecte des ress... Ajouter un attribut "temps alloué au sprint" par p...	112 widget sprint pla... comme isz	110 Impression plan d... avec la liste des taches leur état	11	40	

Le backlog de produit est la liste des futures réalisations de l'équipe.

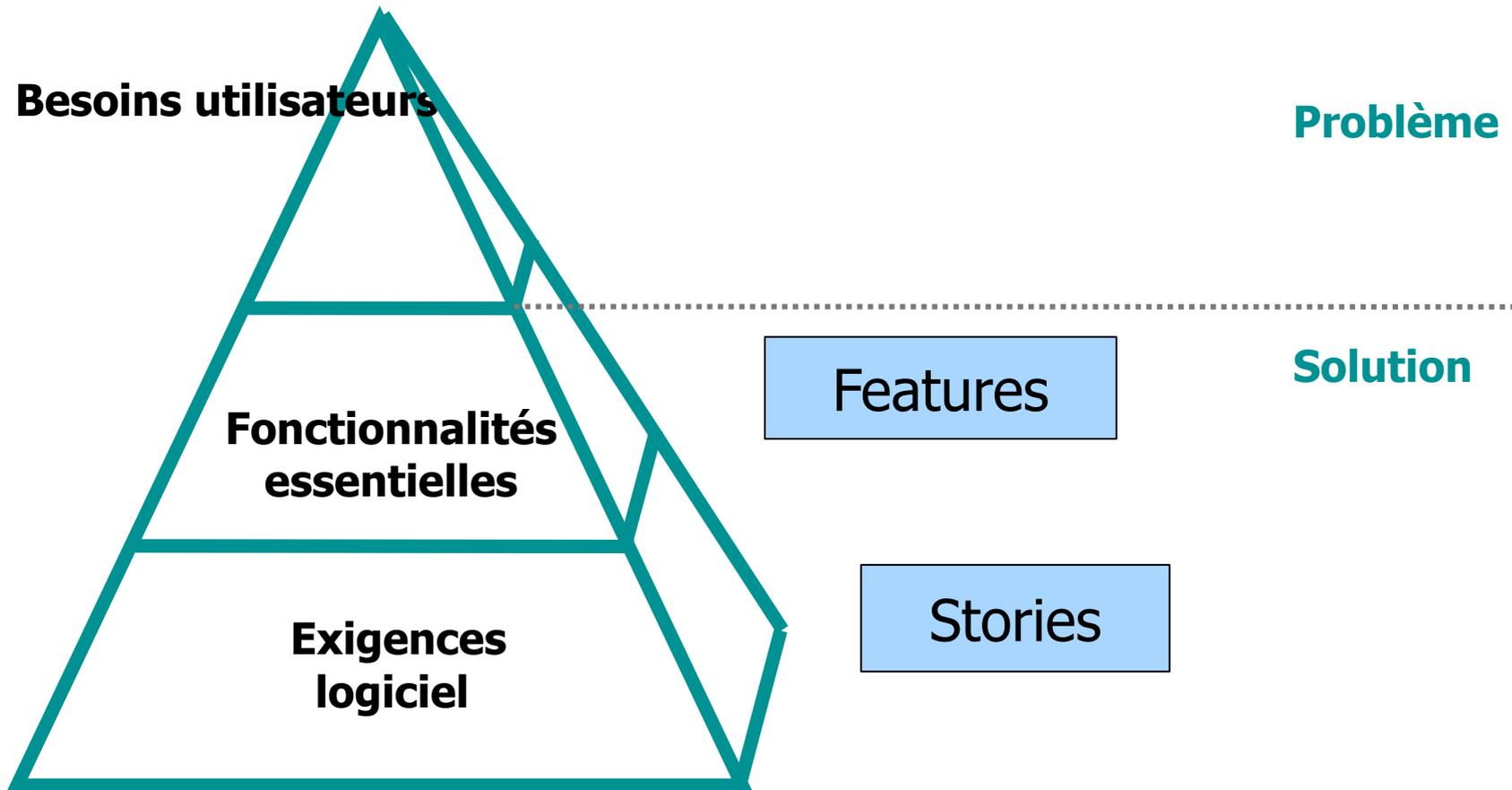


De la vision aux stories

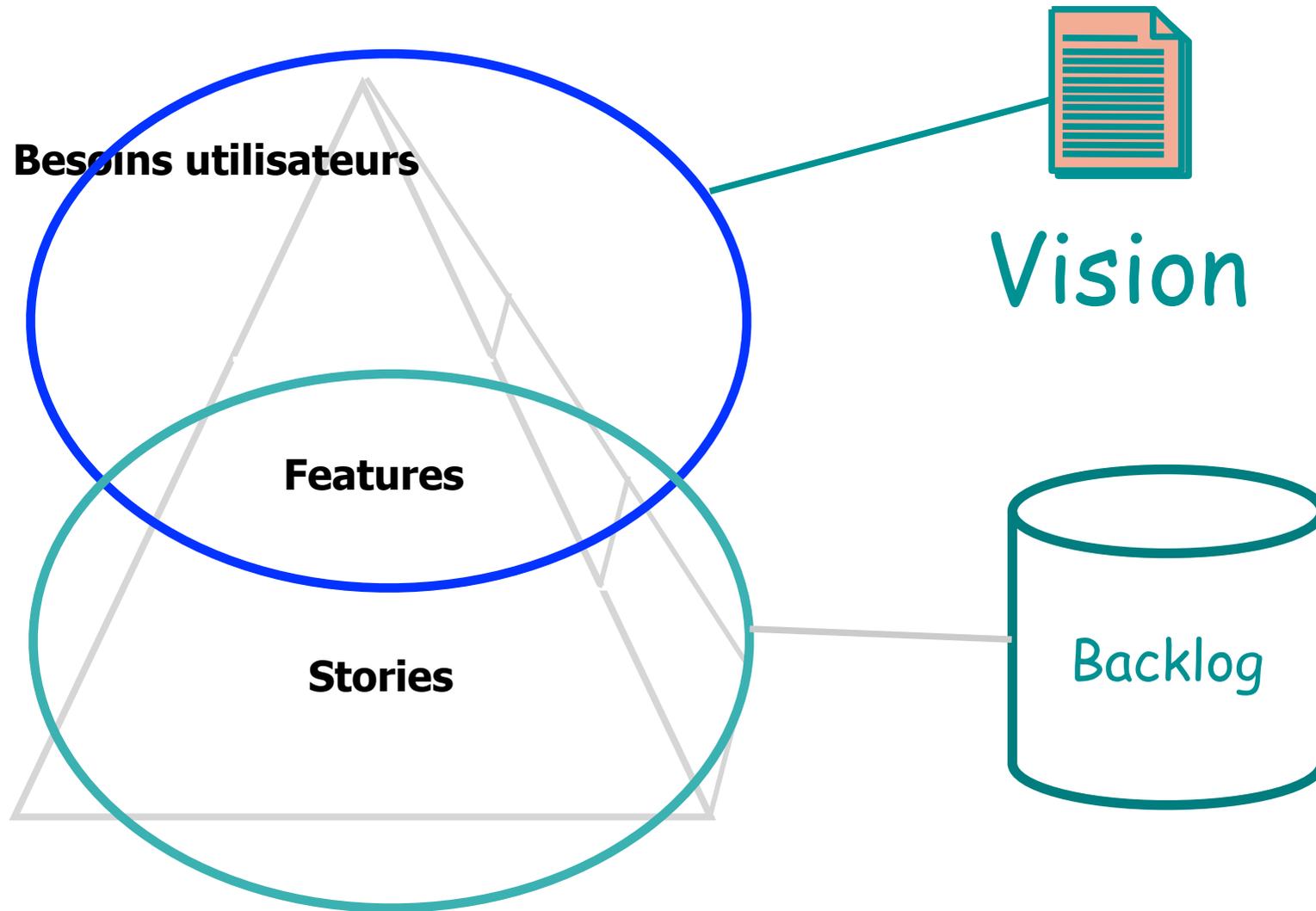
13



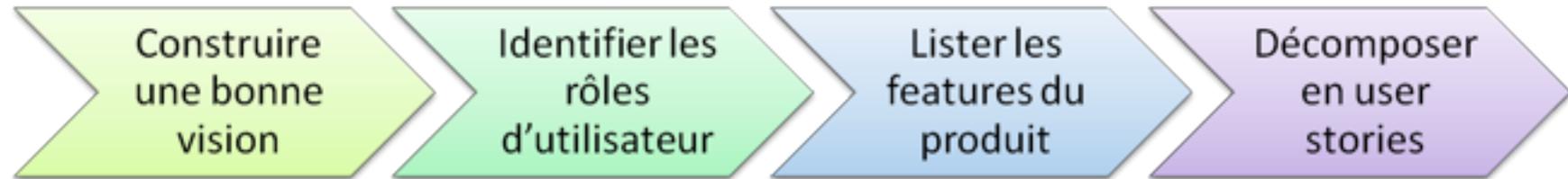
Vocabulaire



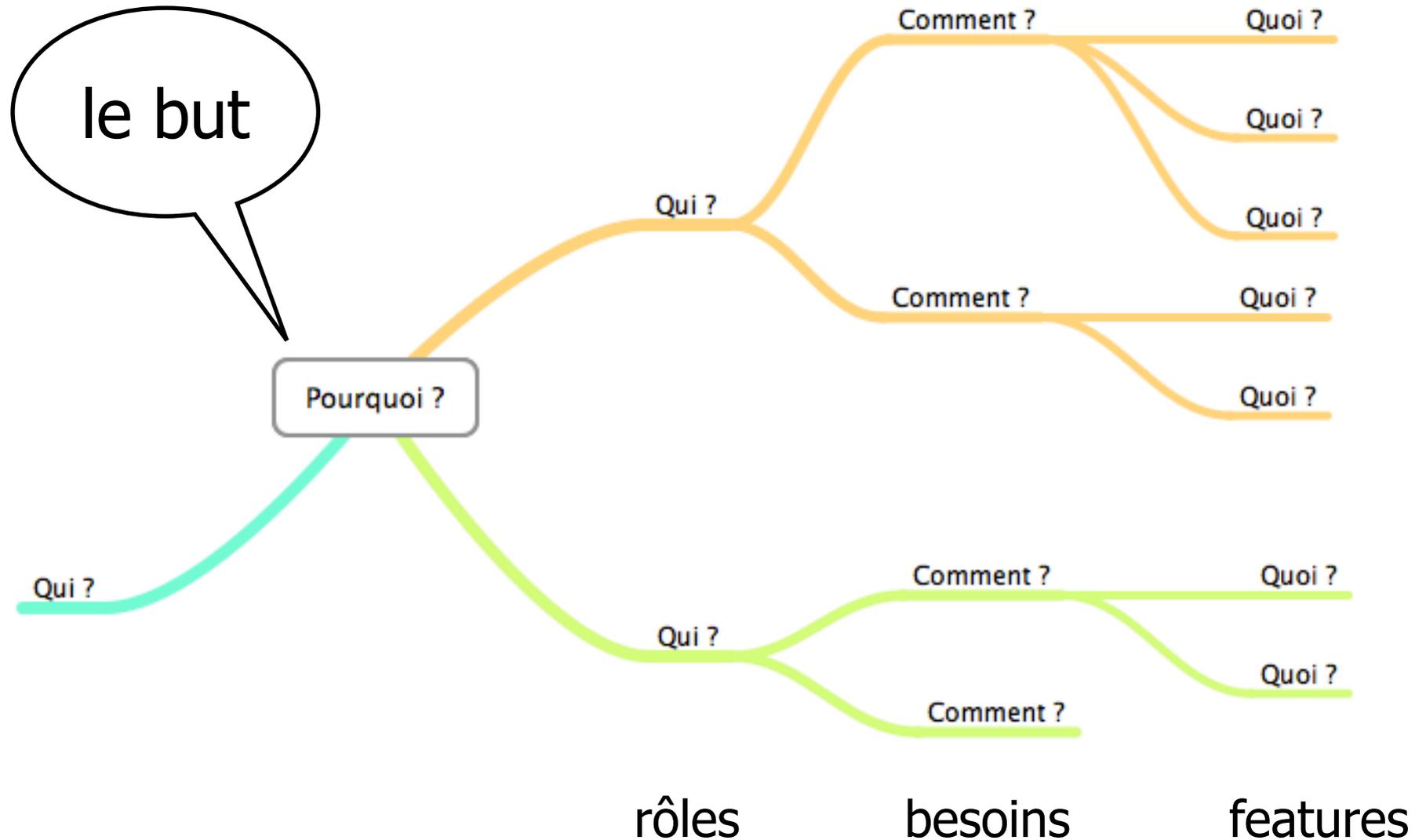
Artefacts agiles



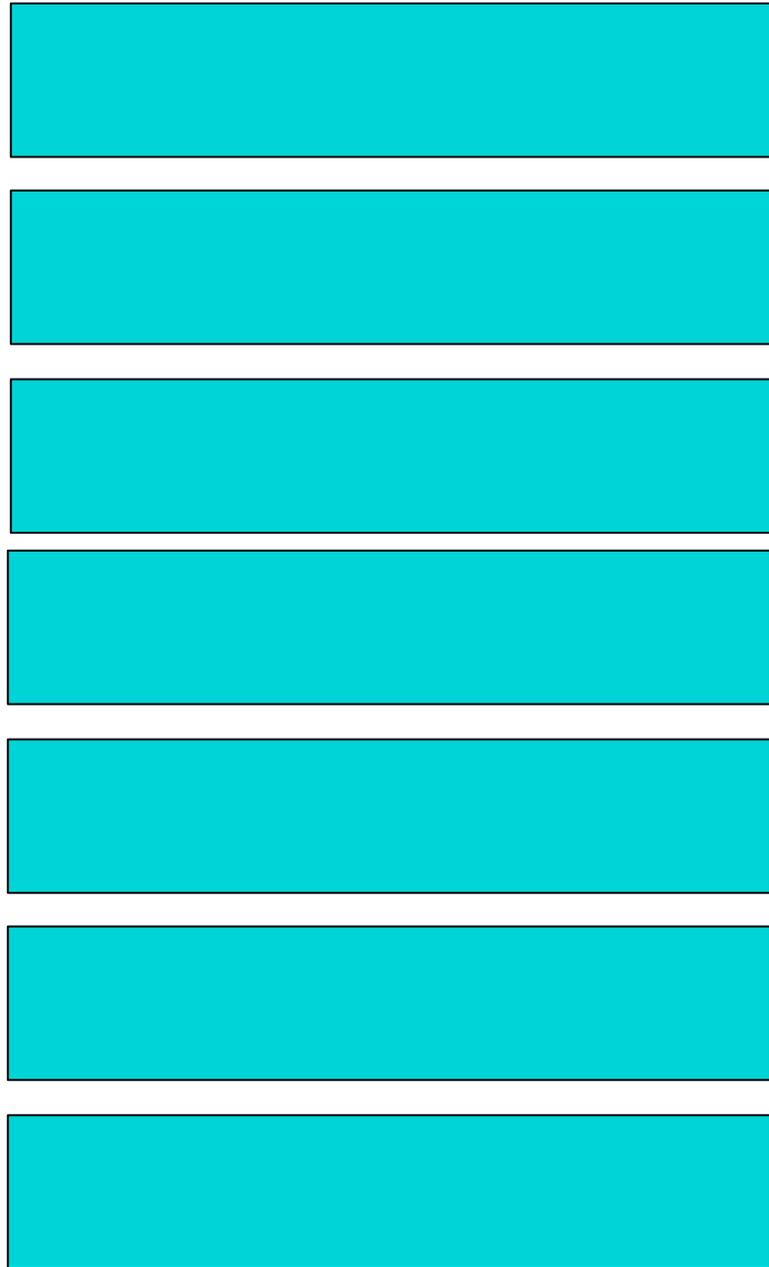
De la vision aux stories



Carte des effets



Backlog avec features



Une feature est une capacité ou un service de haut niveau rendu par le logiciel qui répond à un besoin utilisateur



Qu'est-ce qu'une user story ?

Un petit morceau fonctionnel
qui a de la valeur pour le “métier” et
qui peut être fourni en une itération

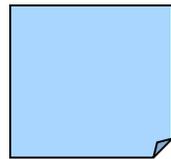
Ce n'est pas une technique de
spécification fonctionnelle



Importance de la communication

Une promesse pour une future conversation

Ron Jeffries
les 3C



Contenu

Carte

Conversation

Confirmation



Formuler une story

En tant que <rôle>,
je peux <intention >
afin de <justification du
point de vue métier>

En tant que opérateur
je crée un compte pour
un client

En tant que commercial
je recherche un client

En tant que dirigeant
je consulte le
récapitulatif d'un client



Exemples

En tant que opérateur
je crée un compte pour
un client

En tant que dirigeant
je consulte le
récapitulatif d'un client

En tant que commercial
je recherche un client



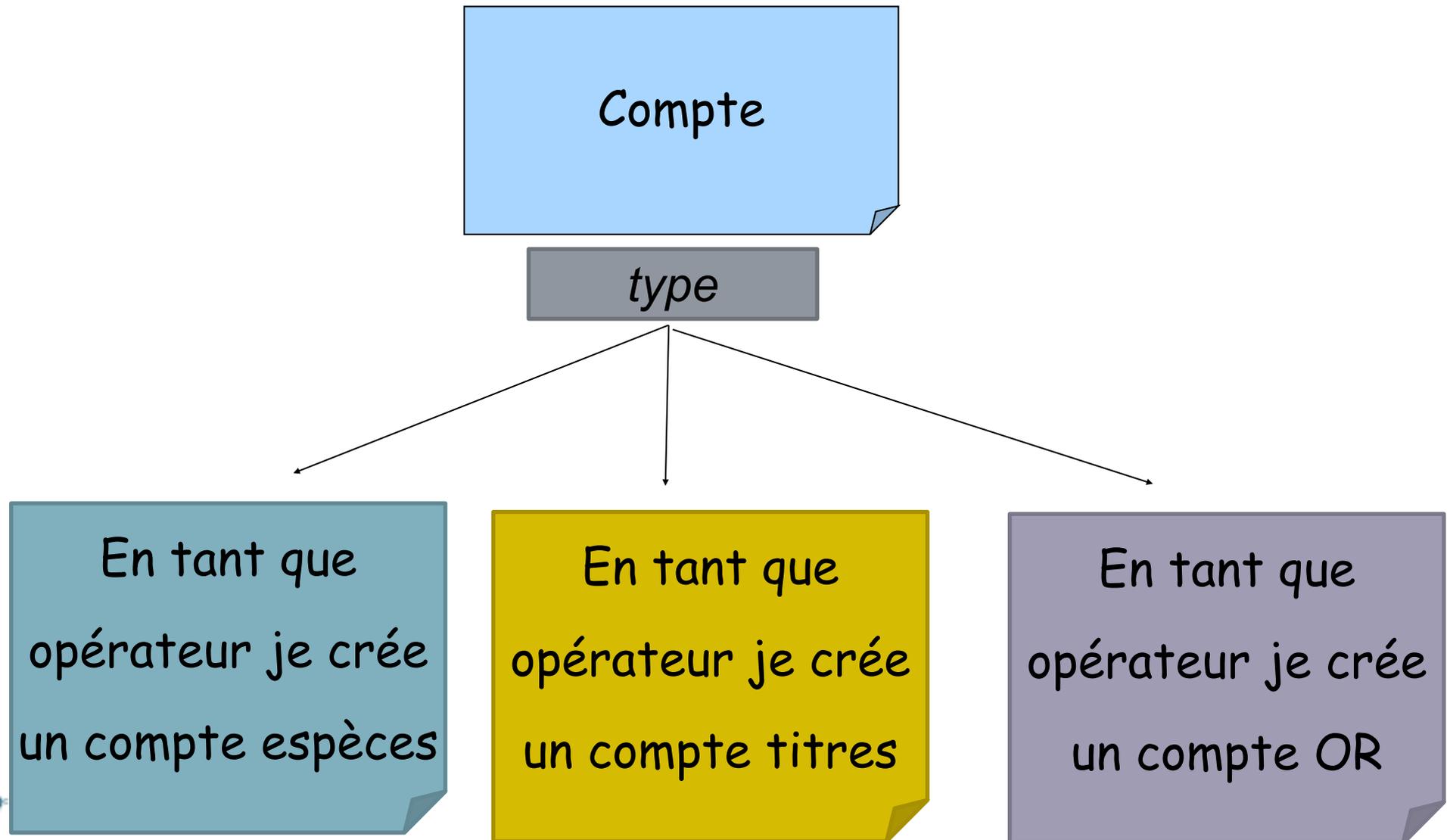
Story élémentaire

En tant qu'opérateur je
crée un compte pour un
client

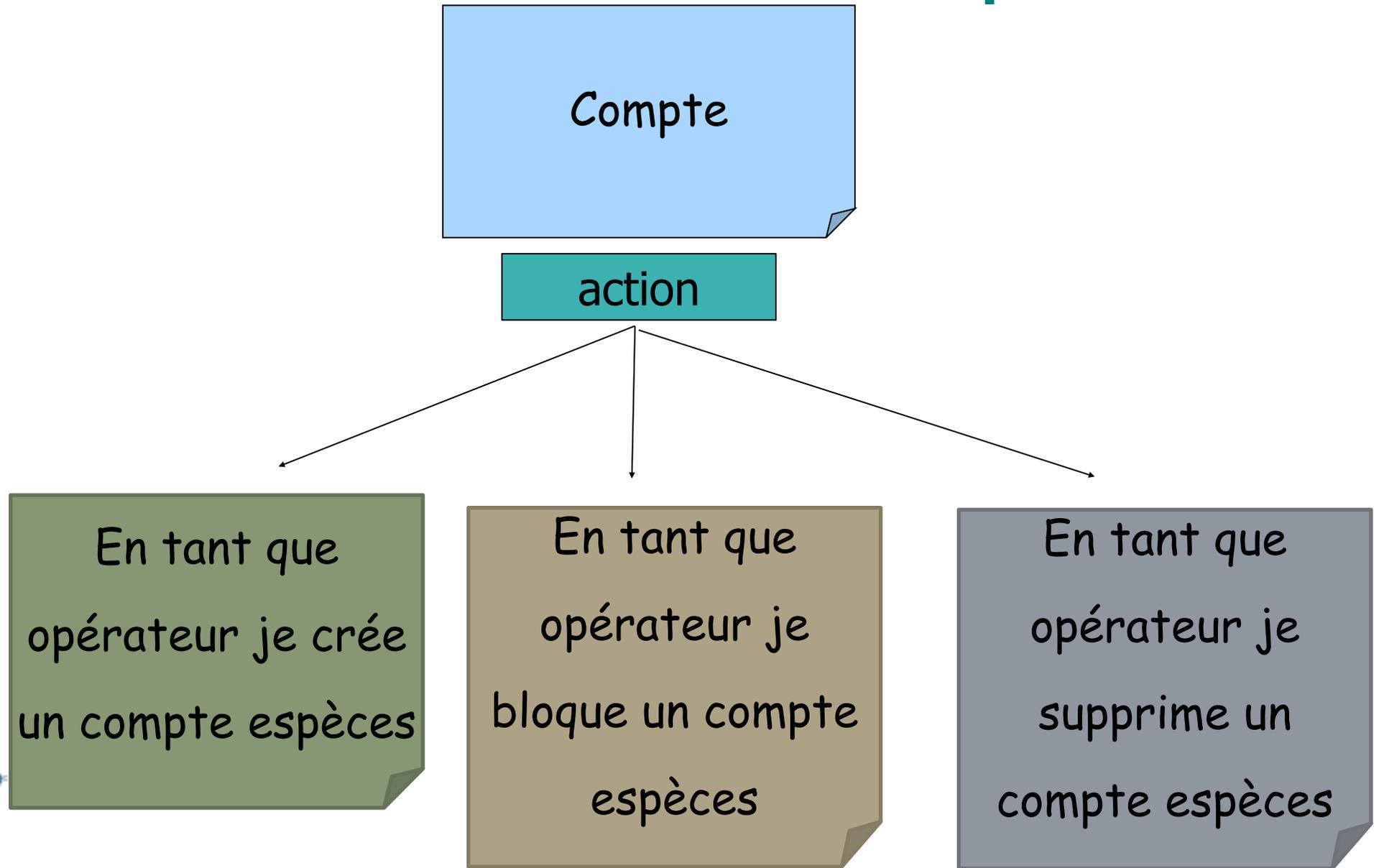
- ▶ Y a-t-il plusieurs types de compte ?

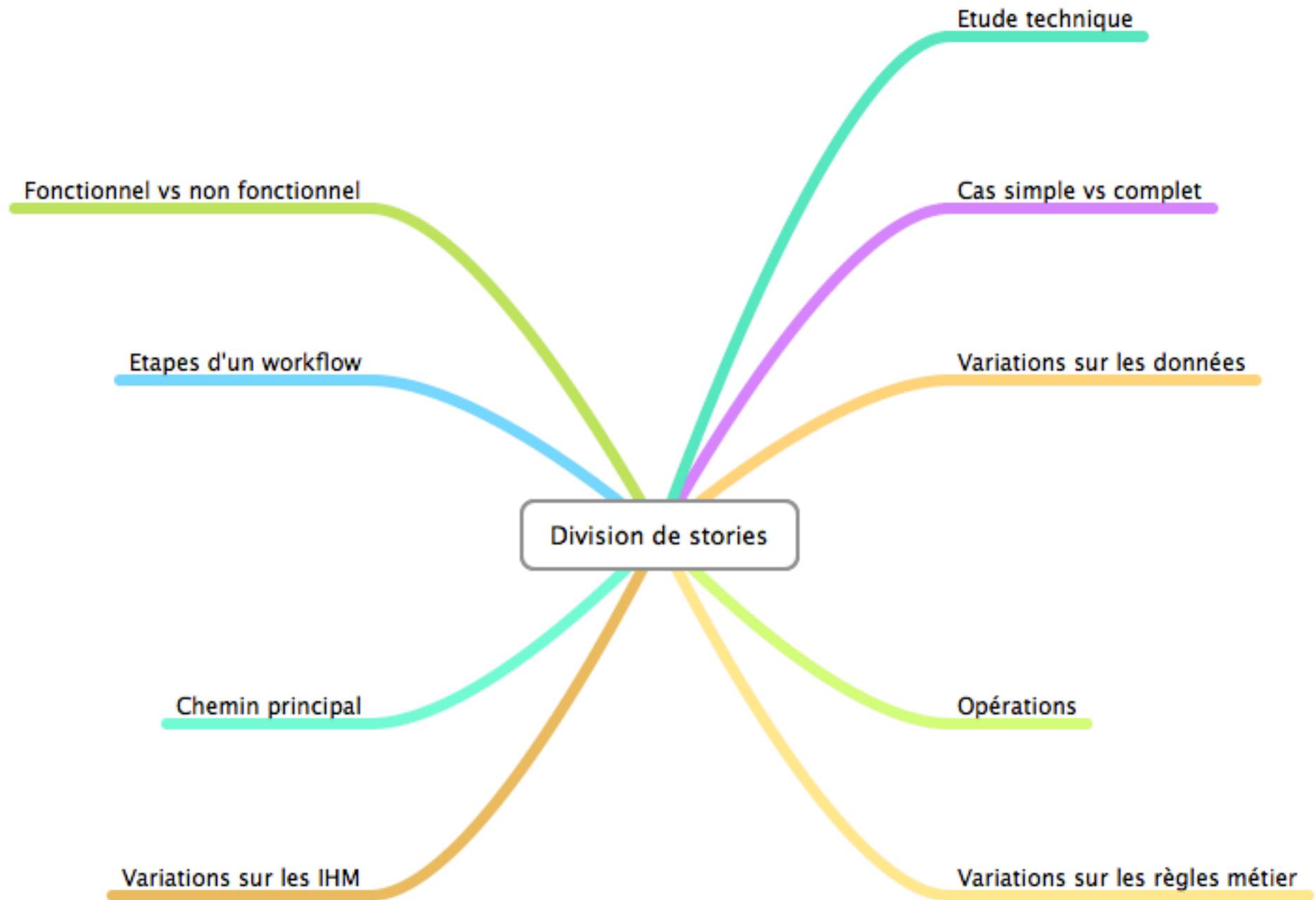


Décomposition selon les données



Décomposition selon les opérations





Où sont les détails ?

En tant qu'opérateur
je crée un compte
espèces pour un client

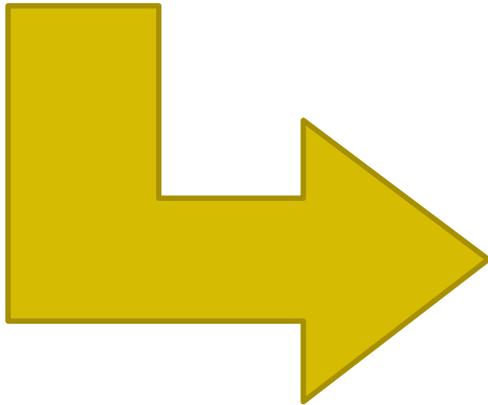
- ▶ Puis-je créer un compte pour tous les clients ?
- ▶ Si je ne suis pas affecté à ce client ?
- ▶ Y a t-il une limite dans le nombre de comptes ?
- ▶ Quels sont les attributs d'un compte ?
- ▶ Ceux obligatoires ?
- ▶ Y a t-il une confirmation ?



Confirmation

En tant que client,
Je peux retirer de l'argent
de mon compte bancaire

- ▶ Chaque story possède des critères d'acceptation



Le solde du
compte doit être
mis à jour avec le
montant retiré par
l'utilisateur



Plus de détail avec les tests

- ▶ Un critère devient un test d'acceptation
 - ▶ Test1 (succès) : le compte est créé avec tous ses attributs
 - ▶ Test2 : le compte n'est pas créé parce qu'il manque le n° de compte
 - ▶ Test3 : il existe déjà un compte espèces
 - ▶ ...

En tant qu'opérateur
je crée un compte
espèces pour un client



Intérêt des stories

Accent mis sur la communication plutôt que l'écrit

Accent mis sur les objectifs des utilisateurs

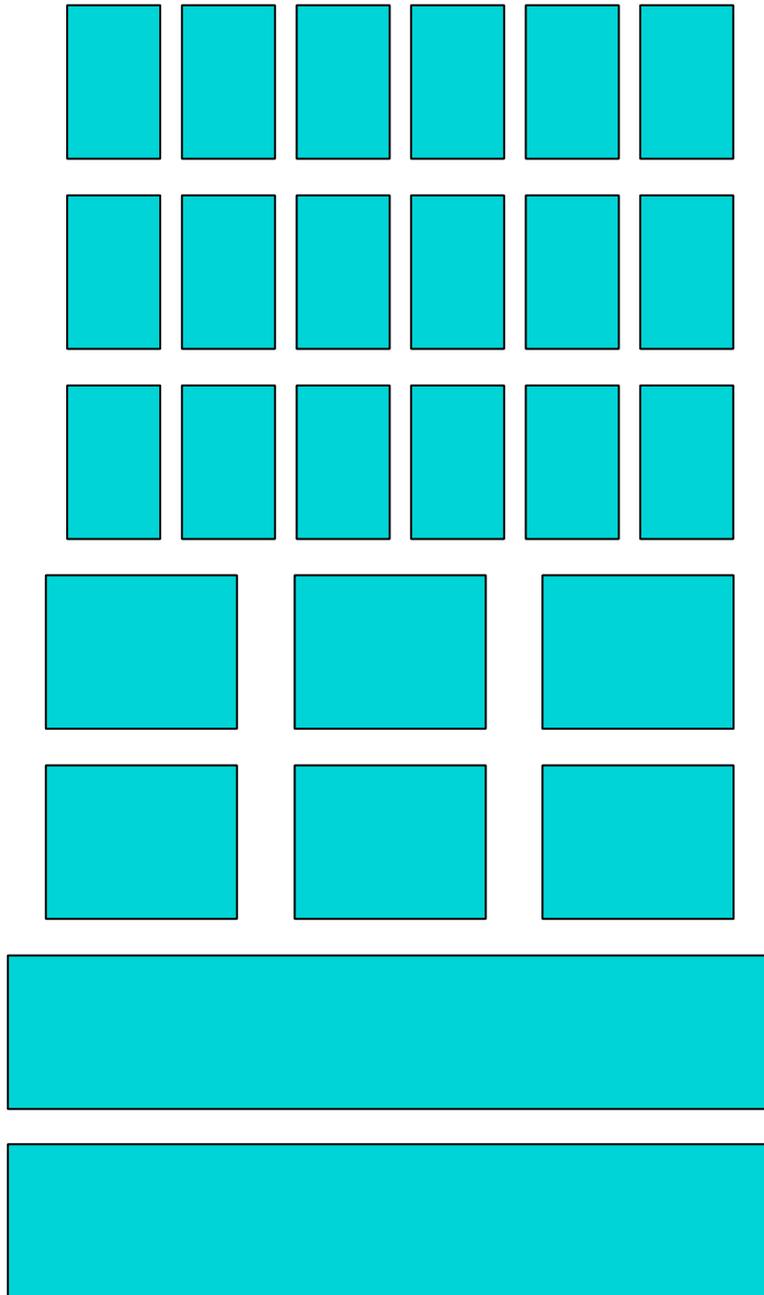
Aussi faciles à comprendre pour les utilisateurs que les développeurs

Permet de connaître l'avancement

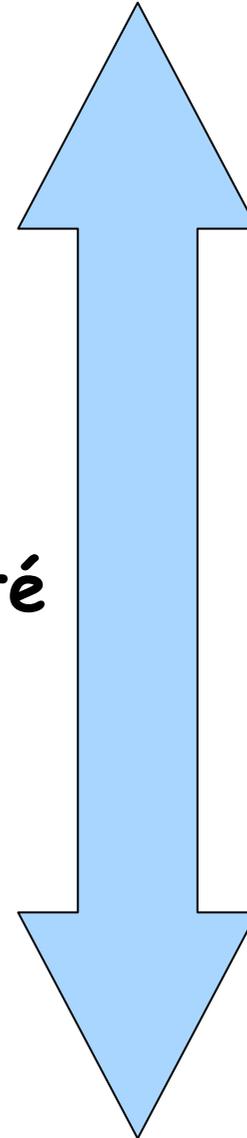
Adaptation à la planification par itérations



Juste à temps



+



Priorité

-

Détailler au
dernier
moment
raisonnable



Story et feature

Story

- Une story apporte un peu de valeur mais pas toujours suffisamment pour être utile
- Finie dans un sprint
- Exemple dans iceScrum : ajouter un commentaire associé à une story

Feature

- Une feature apporte suffisamment de valeur pour être relâchée
- Finie dans une release
- Exemple dans iceScrum : commentaires
- Autres stories associés : supprimer, modifier, répondre
- Test associé à la feature : exporter



La planification de release 6



Objectifs

Répondre à des questions comme :

- Pourrons-nous utiliser la gestion des inscriptions en ligne pour la conférence de mars ?
- Dans combien de temps pouvons-nous espérer diffuser des conférences en streaming ?
- Quel est le budget nécessaire pour développer la version de mars de l'application ?

Pour prendre des décisions



Des croyances erronées

Pas besoin de plan (de release),
nous sommes agiles !



Avec une méthode agile,
le client peut faire des changements tout le temps,
pas possible de planifier !



Renversement

Non Agile

Agile

Contraintes

Exigences

Coût

Planning

**Dirigé par
les prévisions
initiales**

**Dirigé par la
valeur pour
les clients**

Estimations

Coût

Planning

Fonctionnalités



Release et sprint

Plan de release



Tâches	Heures
Faire ci	4
Faire ça	8
Refaire ci	5
Refaire ça	1

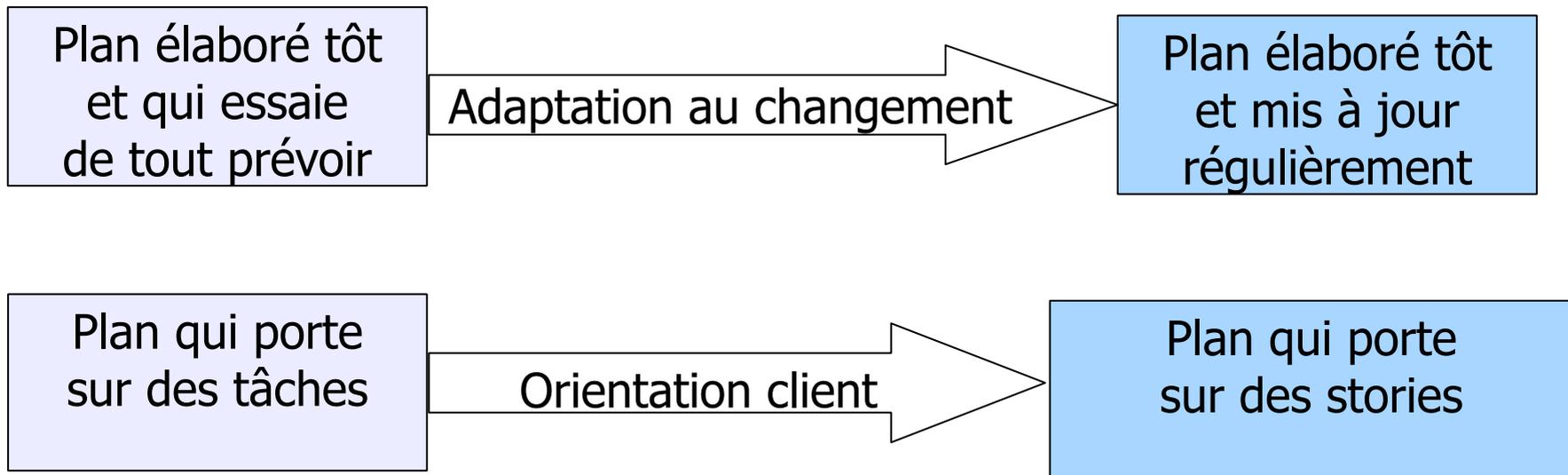
Plan de sprint



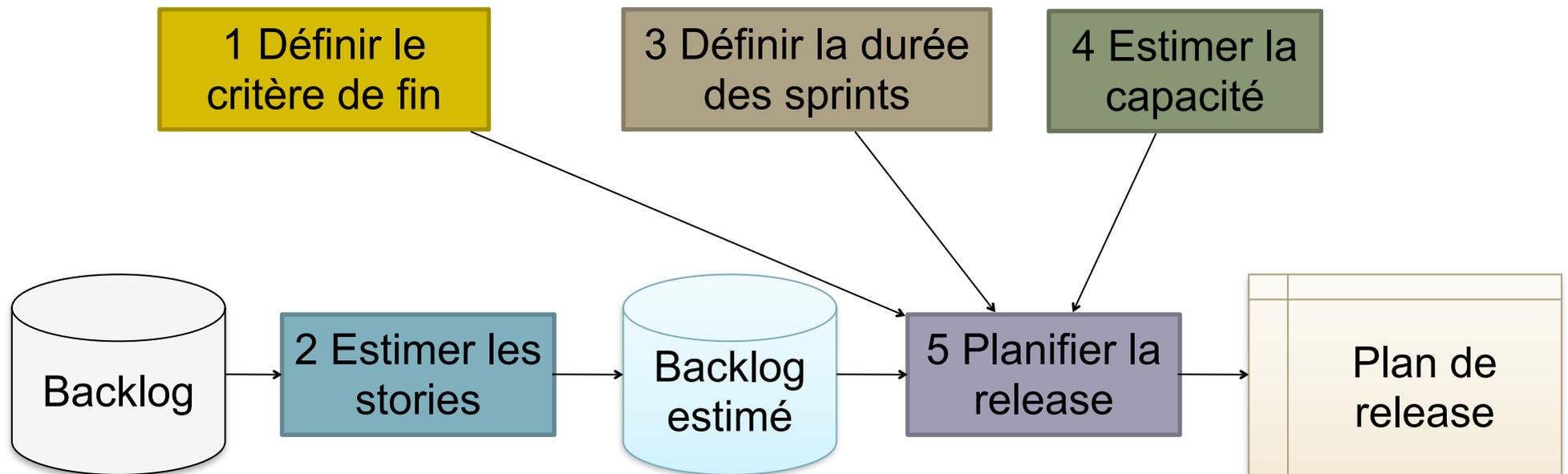
Des changements

Plan projet classique

Plan de release agile

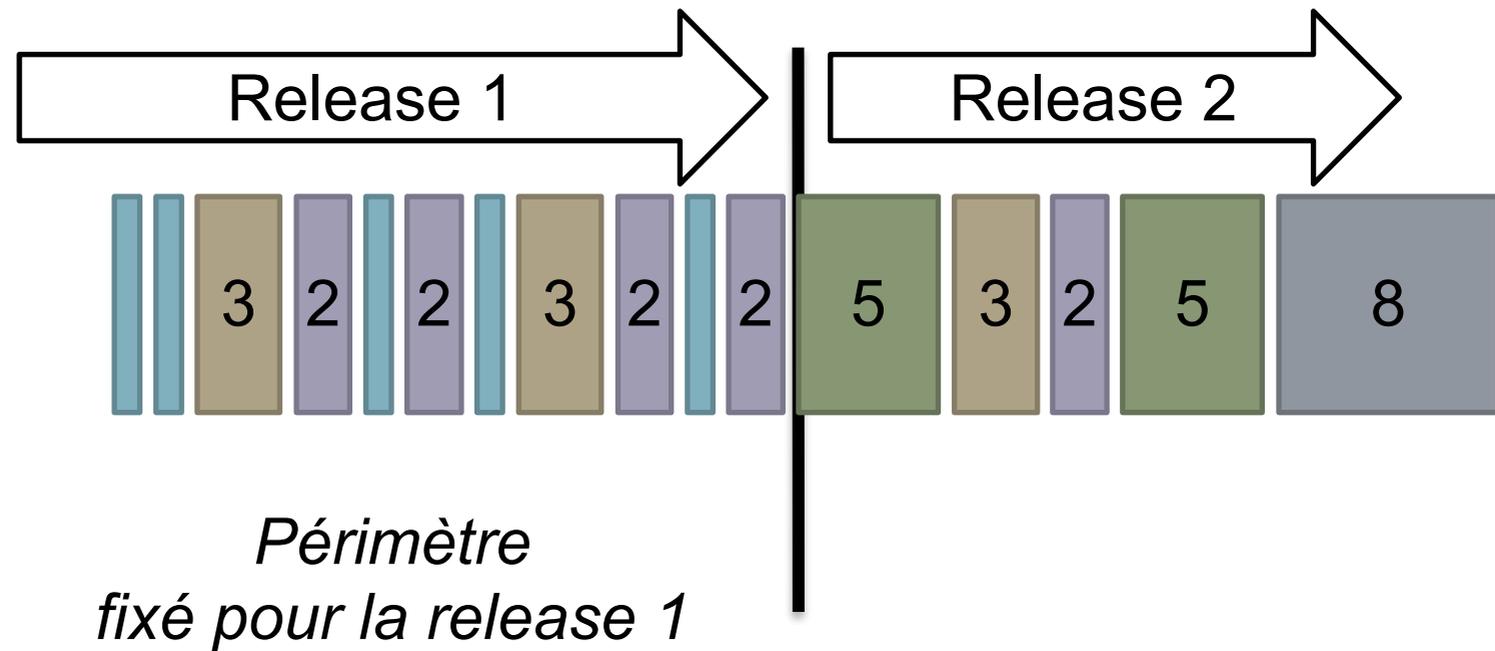


Le processus

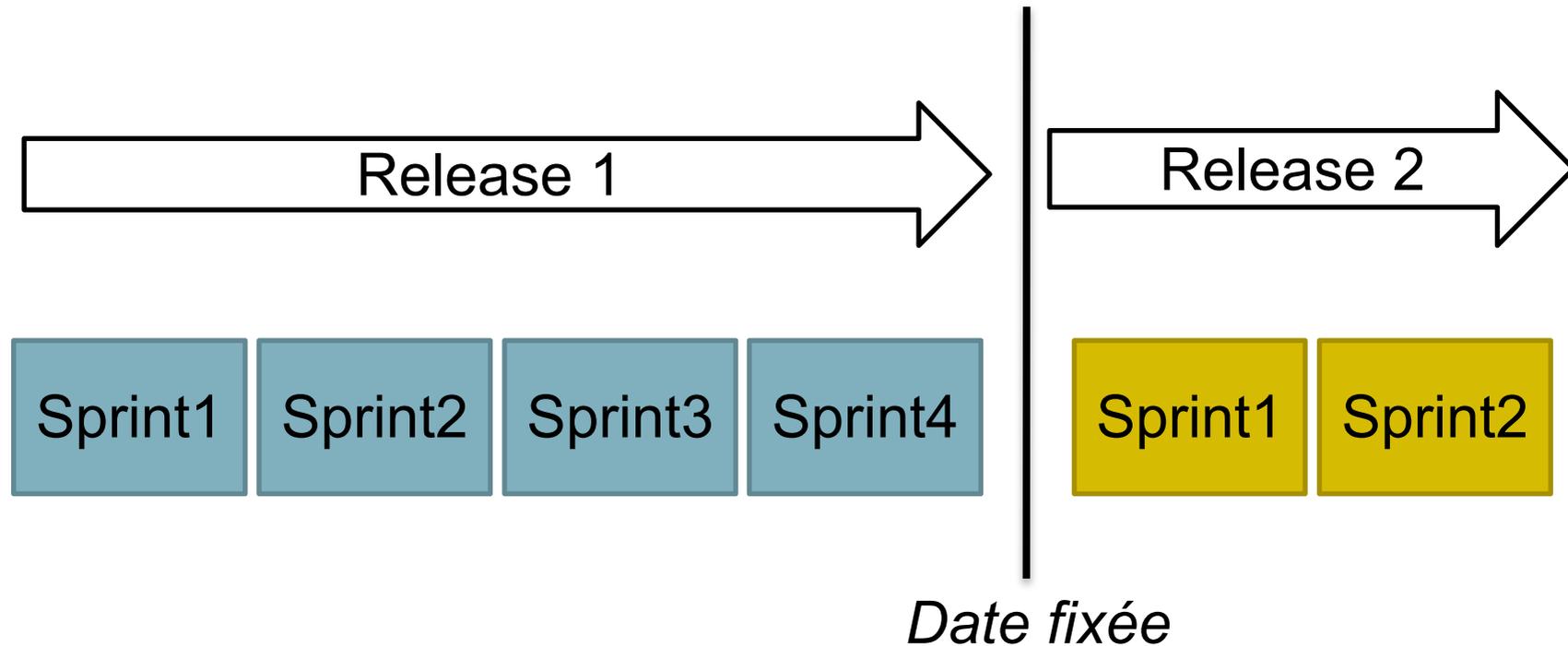


Release à périmètre fixé

backlog



Release à date fixée



Combien de temps pour ranger le bois ?

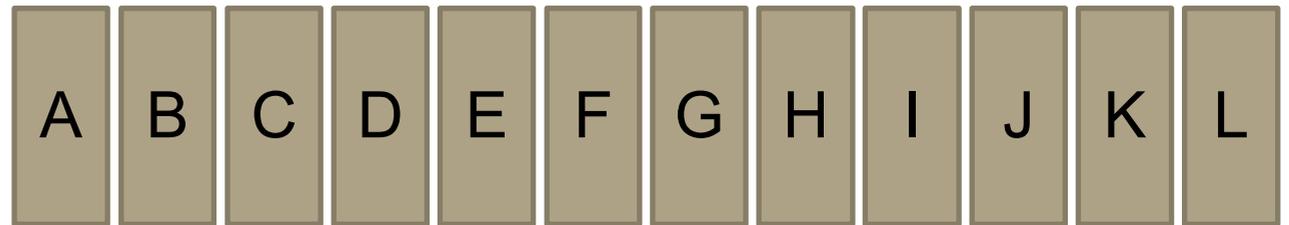


Une estimation n'est pas un engagement

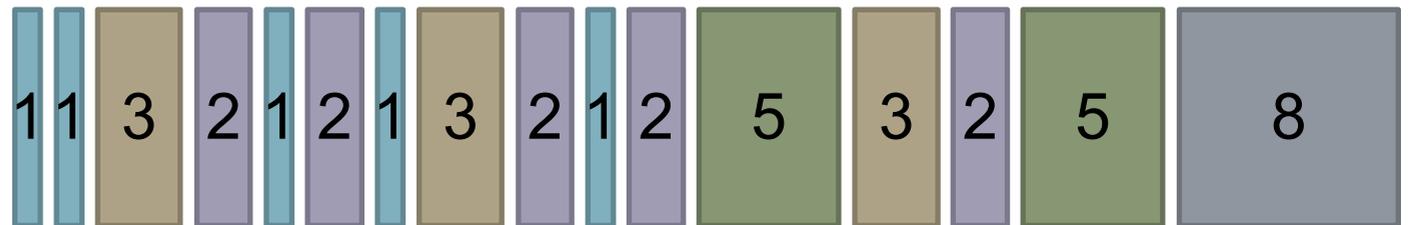
Considérez les estimations faites au début d'un projet comme un budget pour une solution possible, pas comme un engagement sur une solution spécifique

Kent Beck



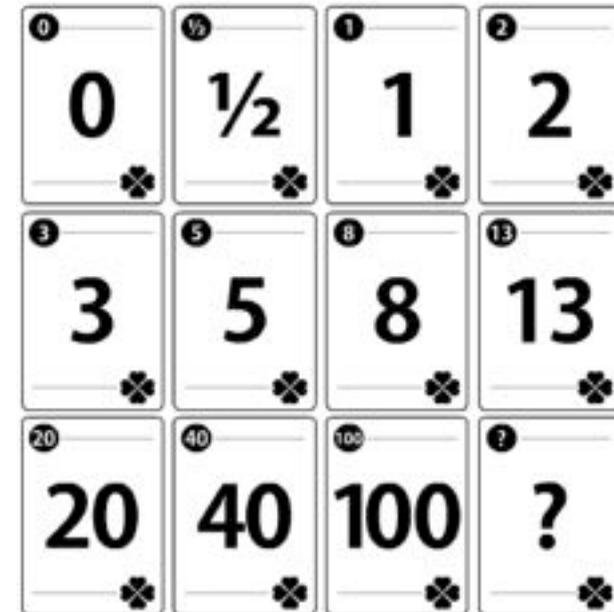


Toutes les stories
ne demandent pas
le même effort



Points

- ▶ Le nombre de points est relatif à l'effort de développement
- ▶ L'estimation de l'effort est influencée par
 - ▶ La complexité (fonctionnelle) de la story
 - ▶ La difficulté (technique) à la réaliser
- ▶ Les points n'ont pas d'unité
 - ▶ On peut se passer de la durée (H/j) grâce au principe de *timebox*
 - ▶ L'estimation se fait par comparaison



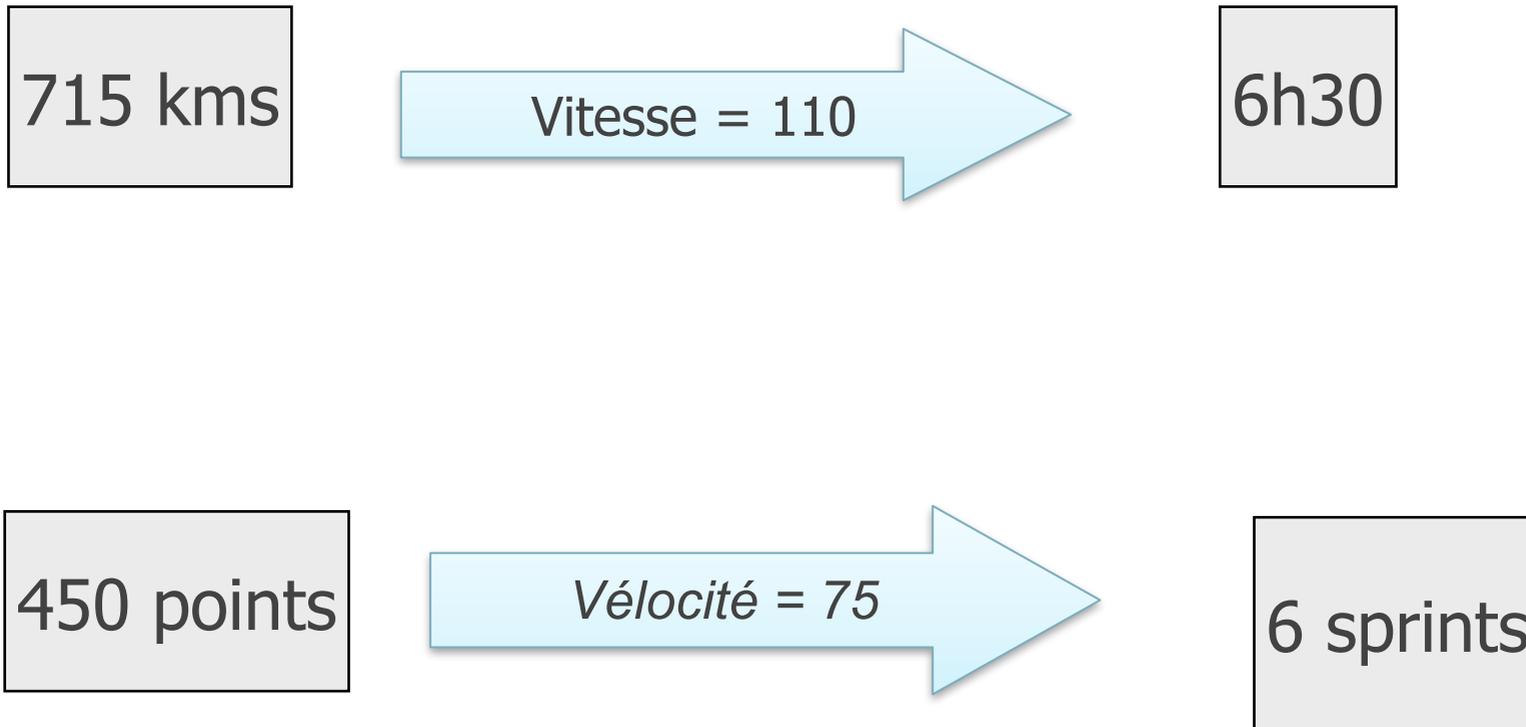
Combien de temps ça prend ?

- ▶ ... pour lire le tome 3 de Millenium ?
- ▶ ... pour aller de Toulouse à Paris en voiture ?
- ▶ ... pour développer une application de gestion des stages ?



Taille et durée

- ▶ Estimer la taille, dériver la durée



Estimation en points

Estimation de l'effort : la durée est dérivée par ce qui est fait dans le sprint

Pas d'unité : estimation par comparaison

1 point



2 points



3 points

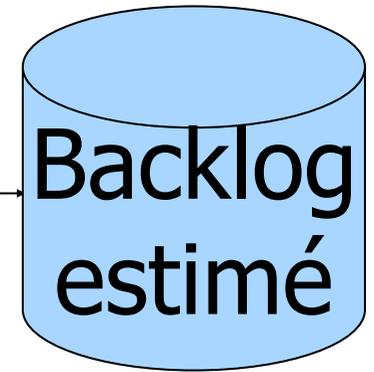
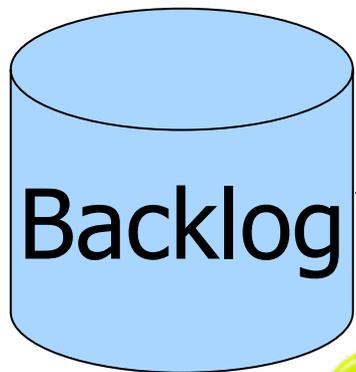


5 points



Planning poker

Session d'estimation collective



Planning poker

Chaque participant reçoit une pile de cartes



Sur chaque carte il y a une valeur possible du coût



Le Product Owner présente la story



Le groupe débat brièvement



Chaque participant présente la carte choisie



Le groupe discute des différences



On continue jusqu'à arriver à une convergence des estimations



Exemple

En tant qu'organisateur
j'annonce une nouvelle
conférence avec sa date

Etalon



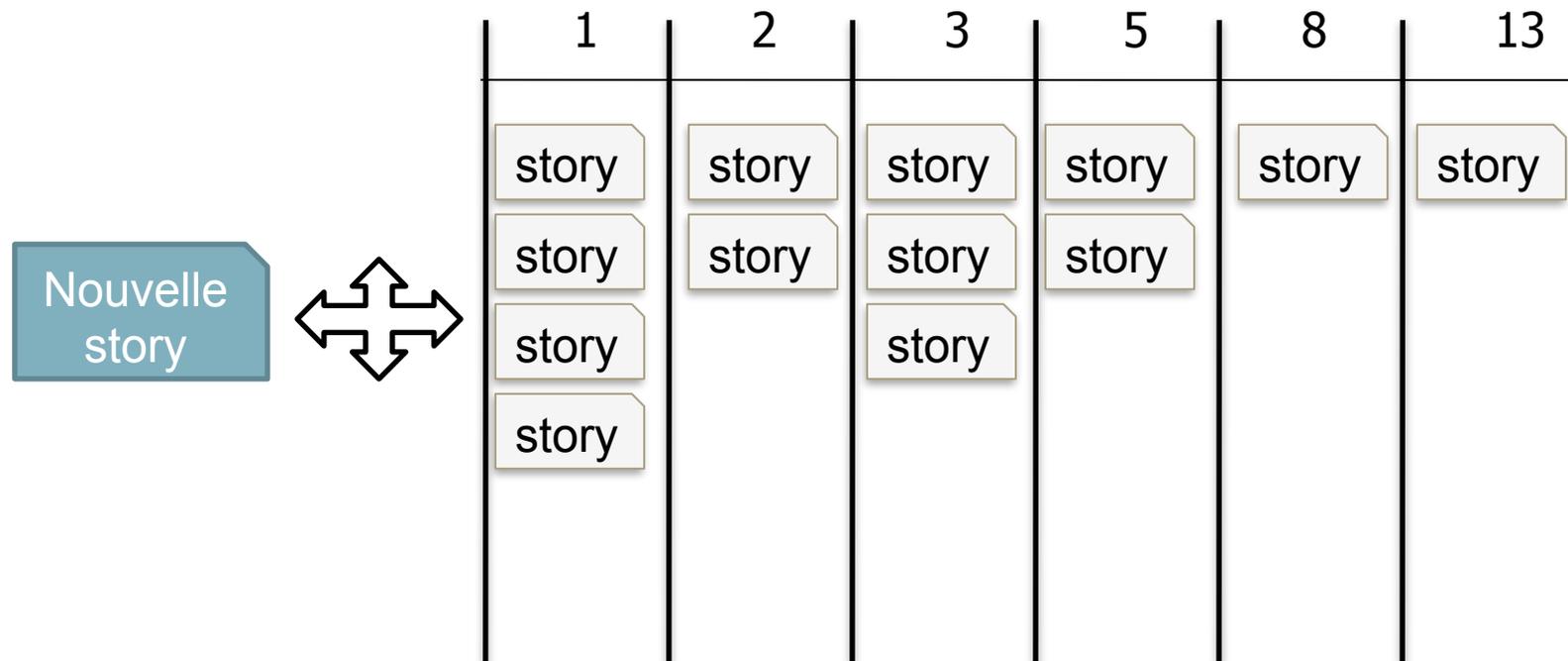
En tant qu'organisateur je
peux connaître les inscrits
à une conférence

	Premier tour	Second tour
Thierry		
Olivier		
Jean-Marie		
David		



Analogie

- ▶ Comparer une story aux autres



Bénéfices

▶ Côté estimation

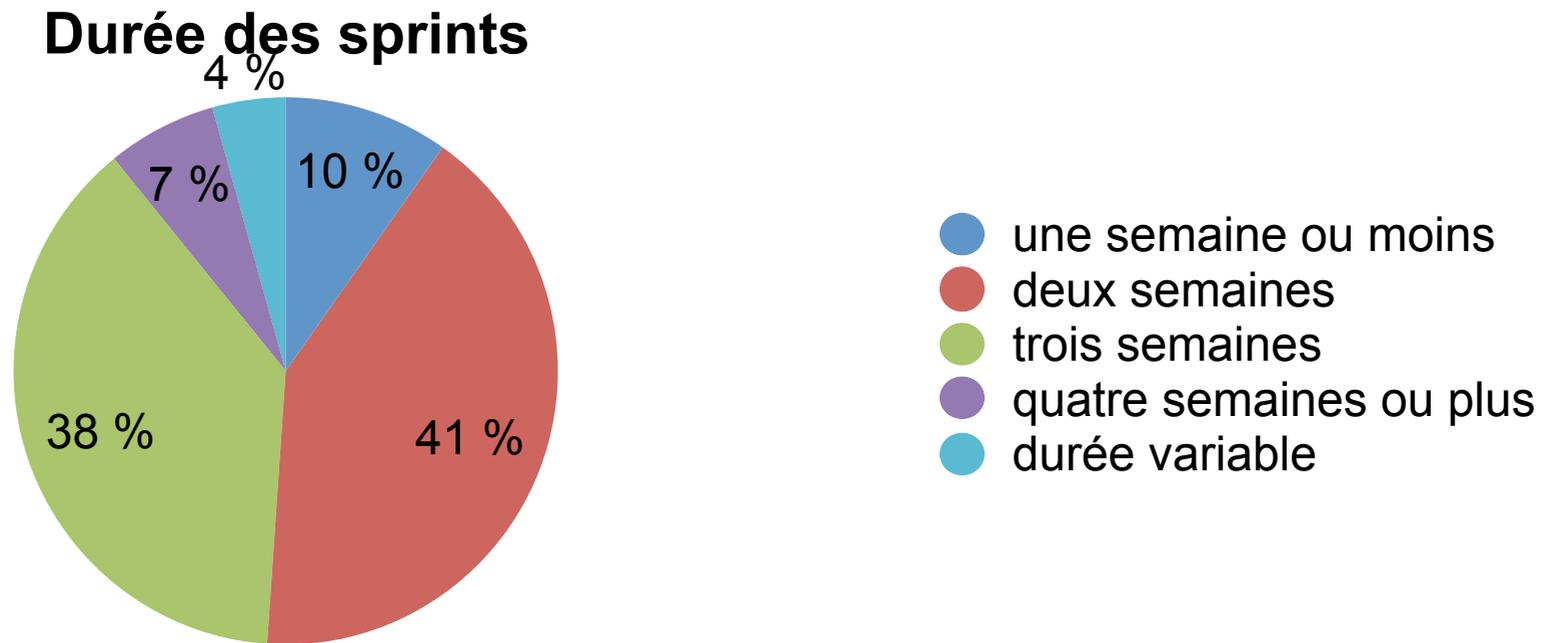
- ▶ Estimation collective
- ▶ Estimation par ceux qui font
- ▶ Rapide pour estimer

▶ Côté équipe

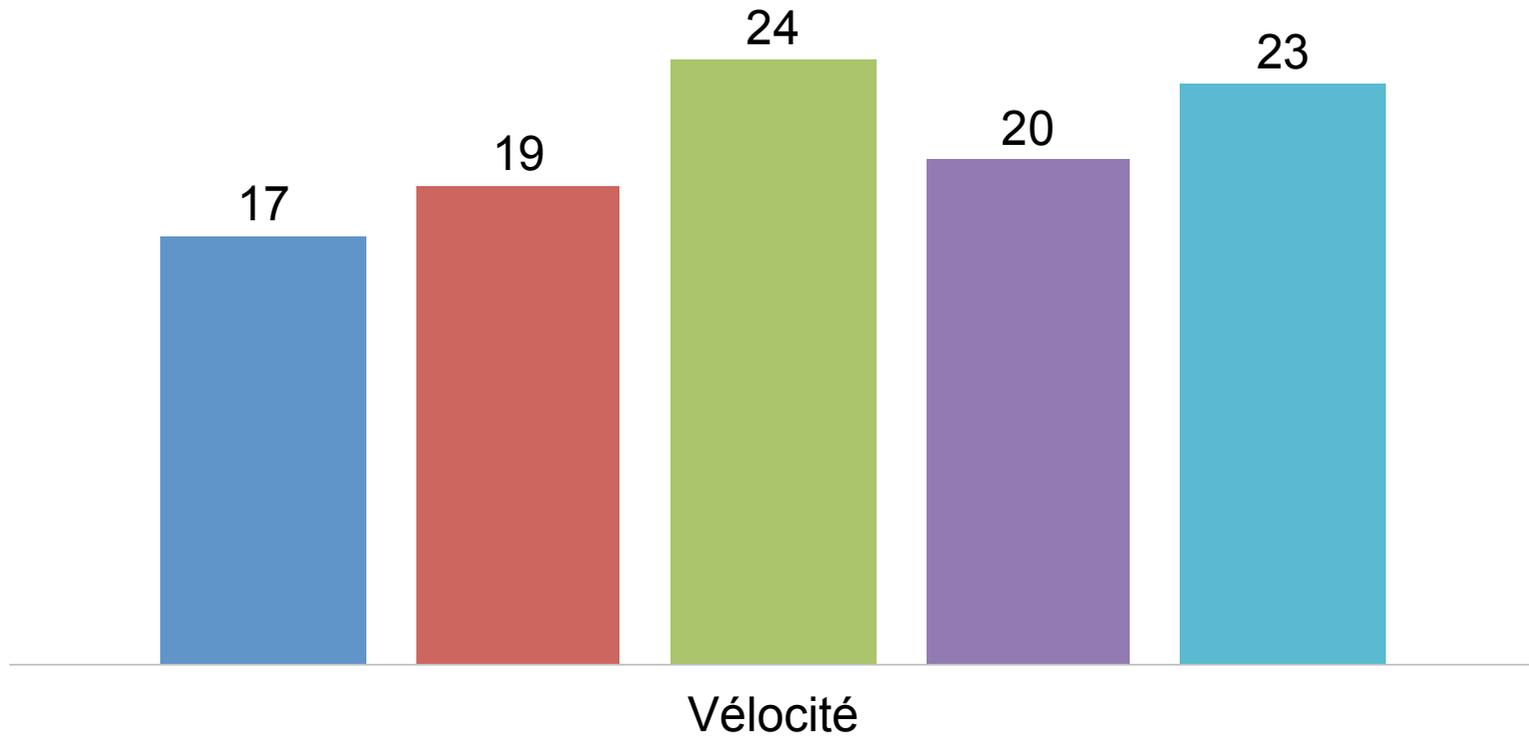
- ▶ Connaissance du produit améliorée
- ▶ Engagement de tous sur les estimations
- ▶ Ludique



Durée d'un sprint



Vélocité mesurée



Vélocité et capacité

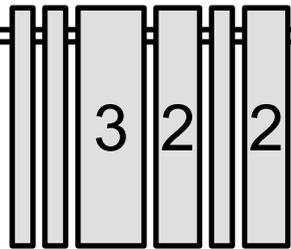
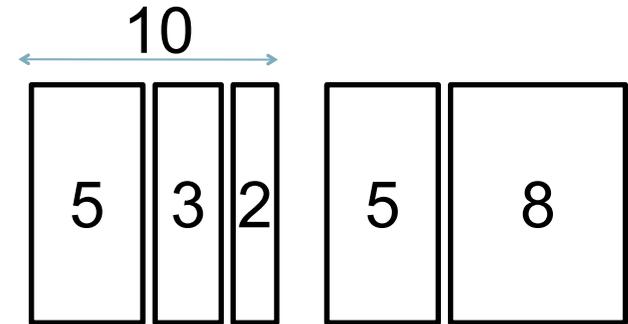
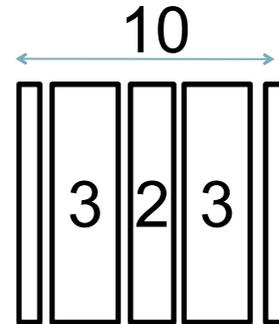
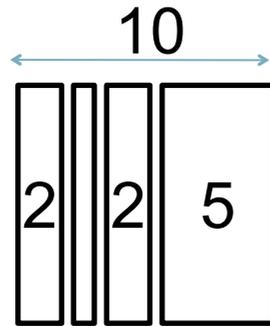
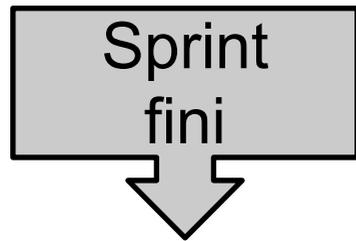
La vélocité est une mesure sur les sprints passés

La capacité est une prévision sur les sprints futurs

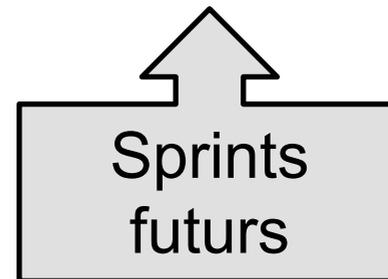
Pour estimer la capacité, le mieux est de se baser sur la vélocité



Association aux sprints



Vélocité = 10



Plan de release

Sprint1

15/11-30/11

Sprint2

Sprint3

Sprint4

2
Story 3

5
Story 1

3
Story 6

8
Story 8

1
Story 4

3
Story 2

5
Story 7

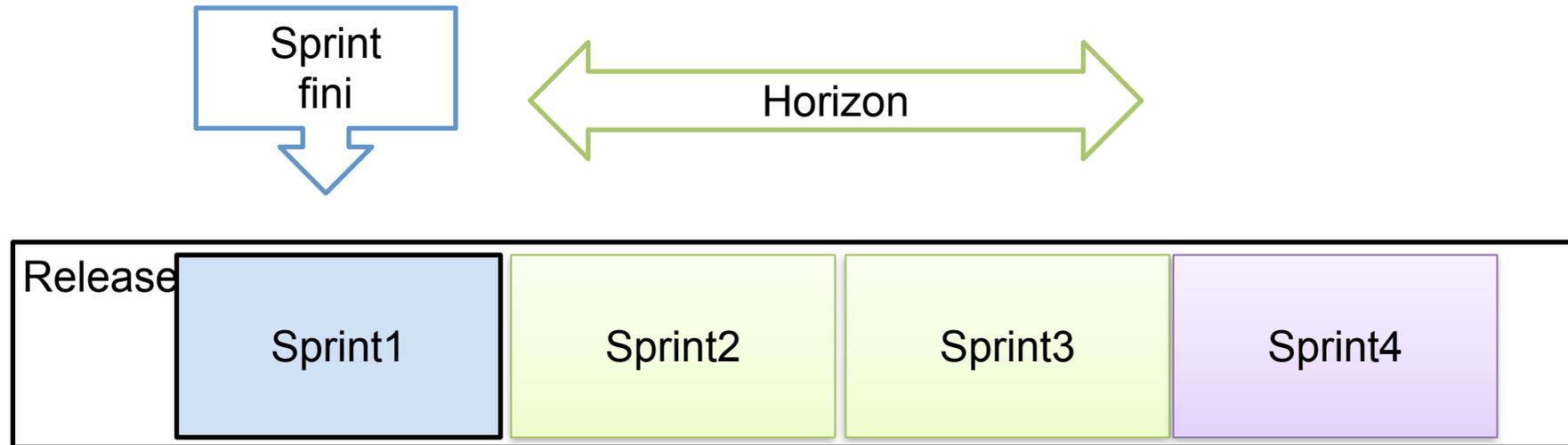
5
Story 5

Date de fin
fixée

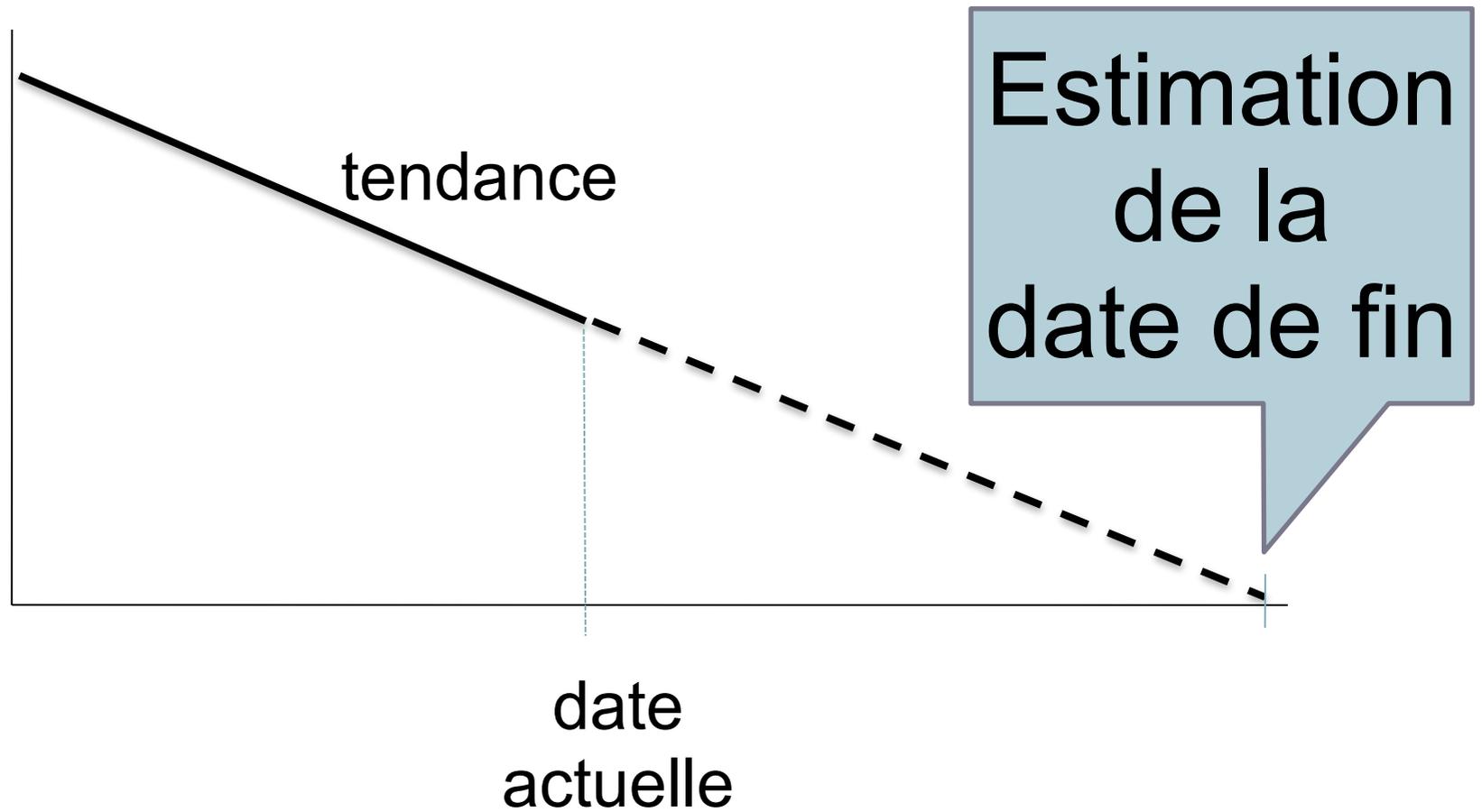
points



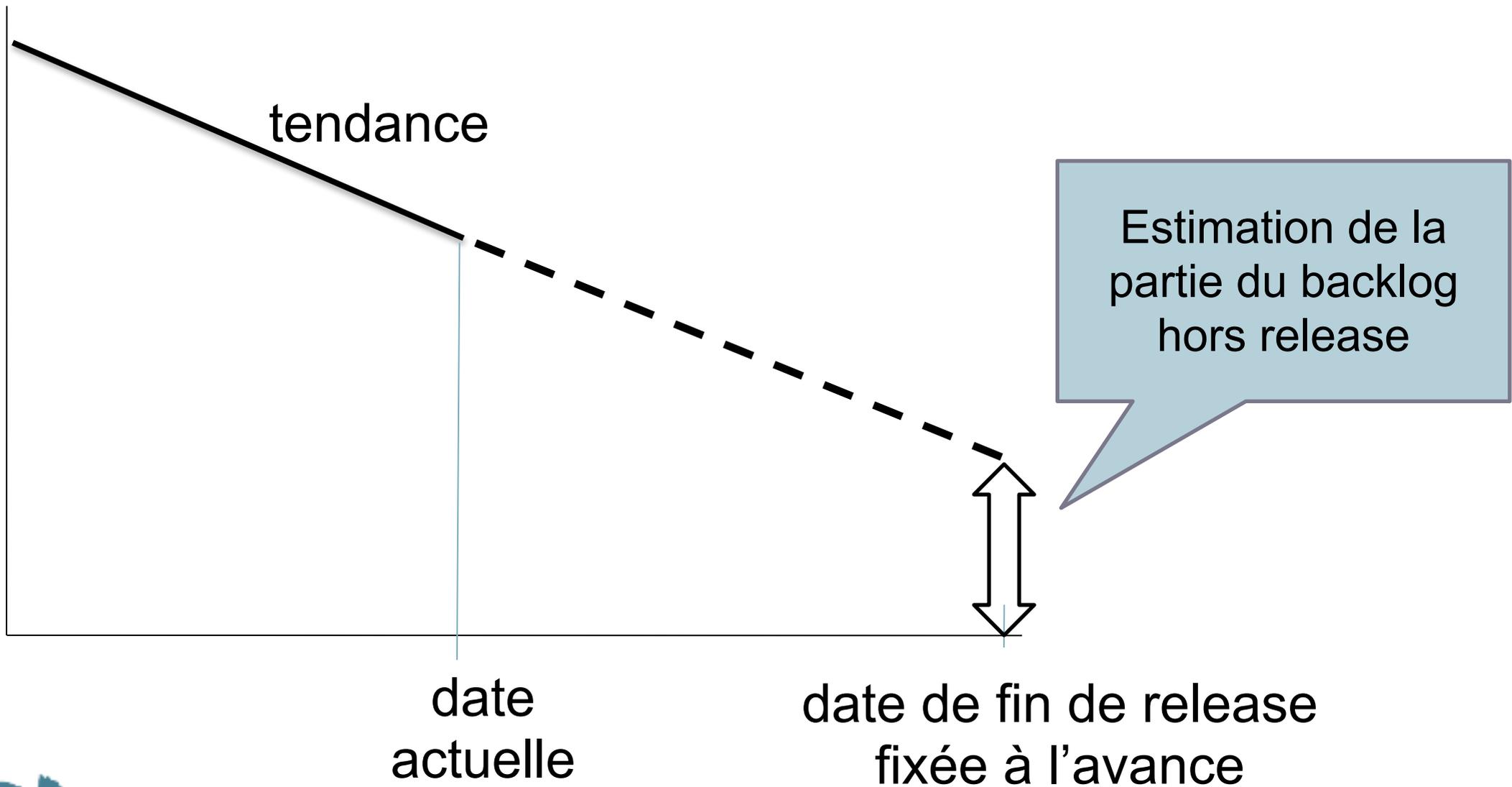
Horizon du plan



Estimation de la date



Estimation du périmètre



Mou pour les incertitudes

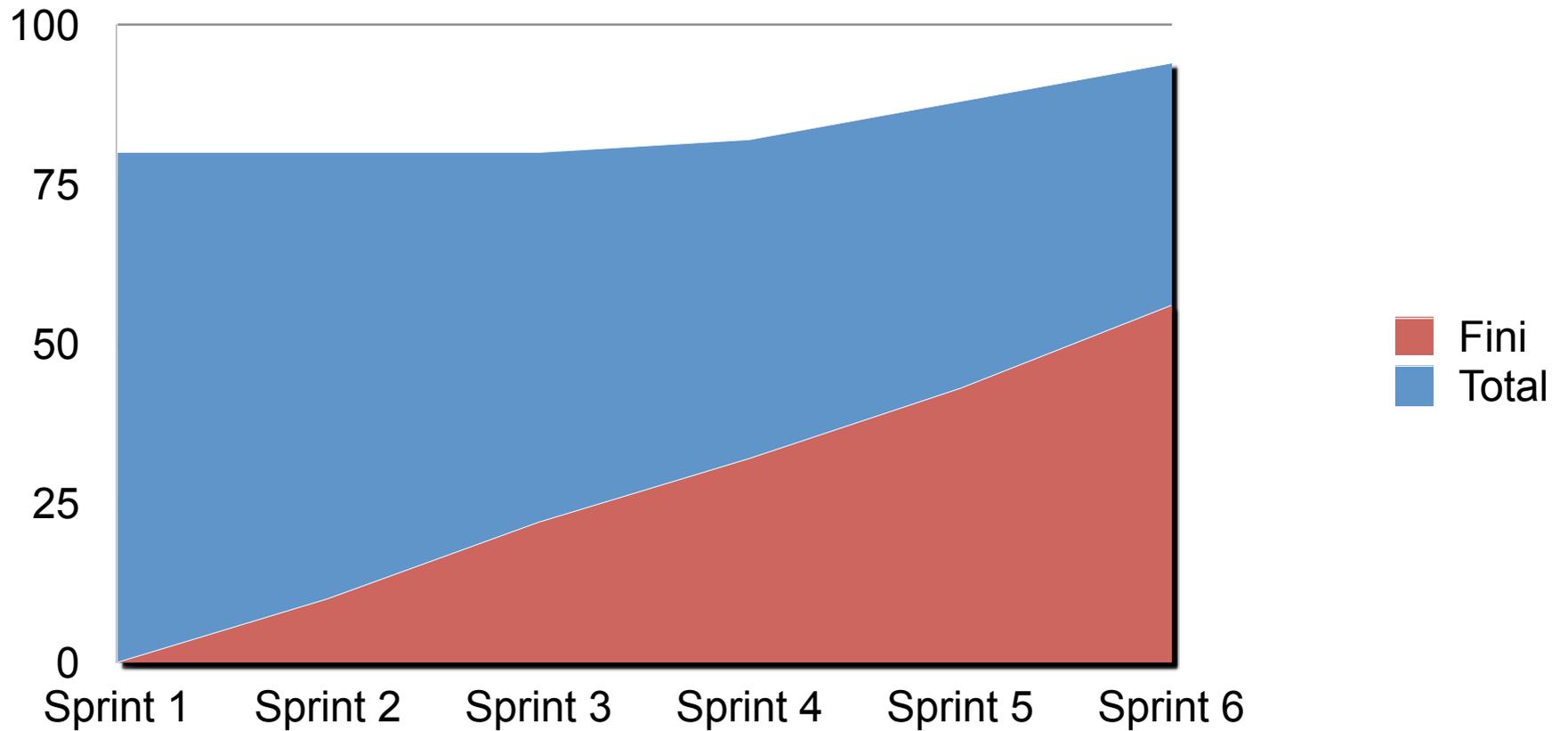
*Incertitude sur
les estimations*



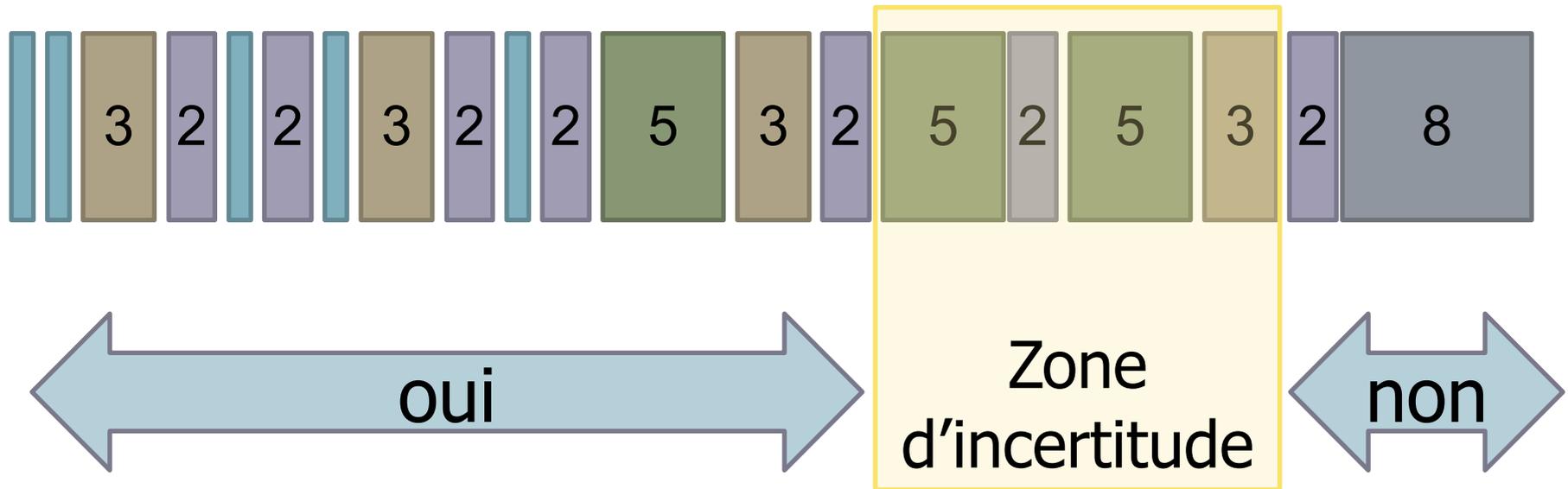
Capacité
de la Release



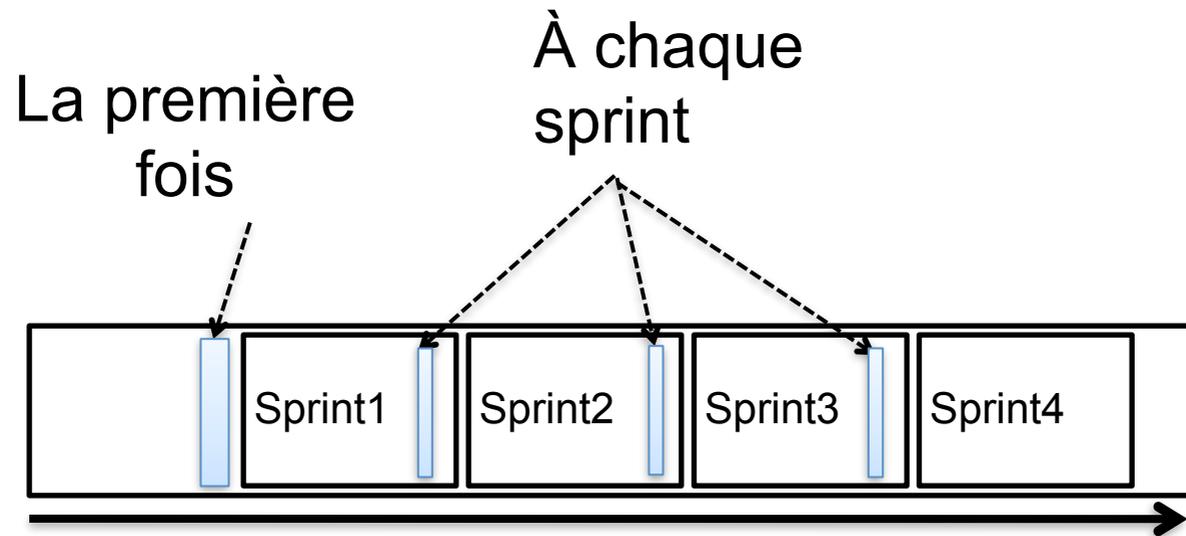
Provision pour feedback



Projection sur le backlog



Quand planifier la release?

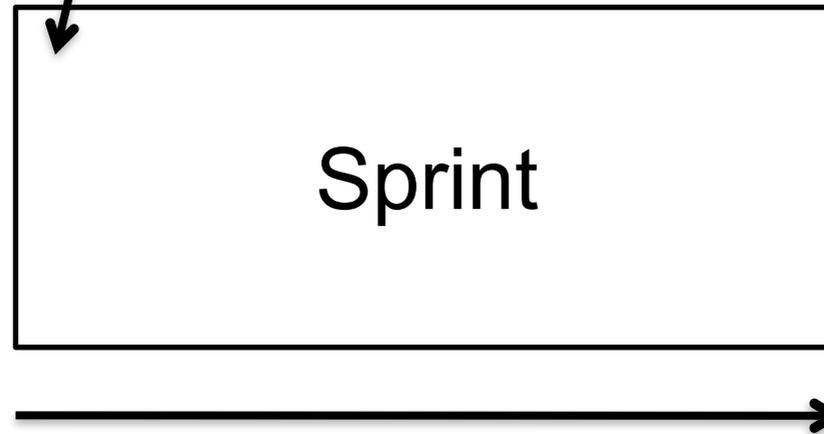


La réunion de planification du sprint 7

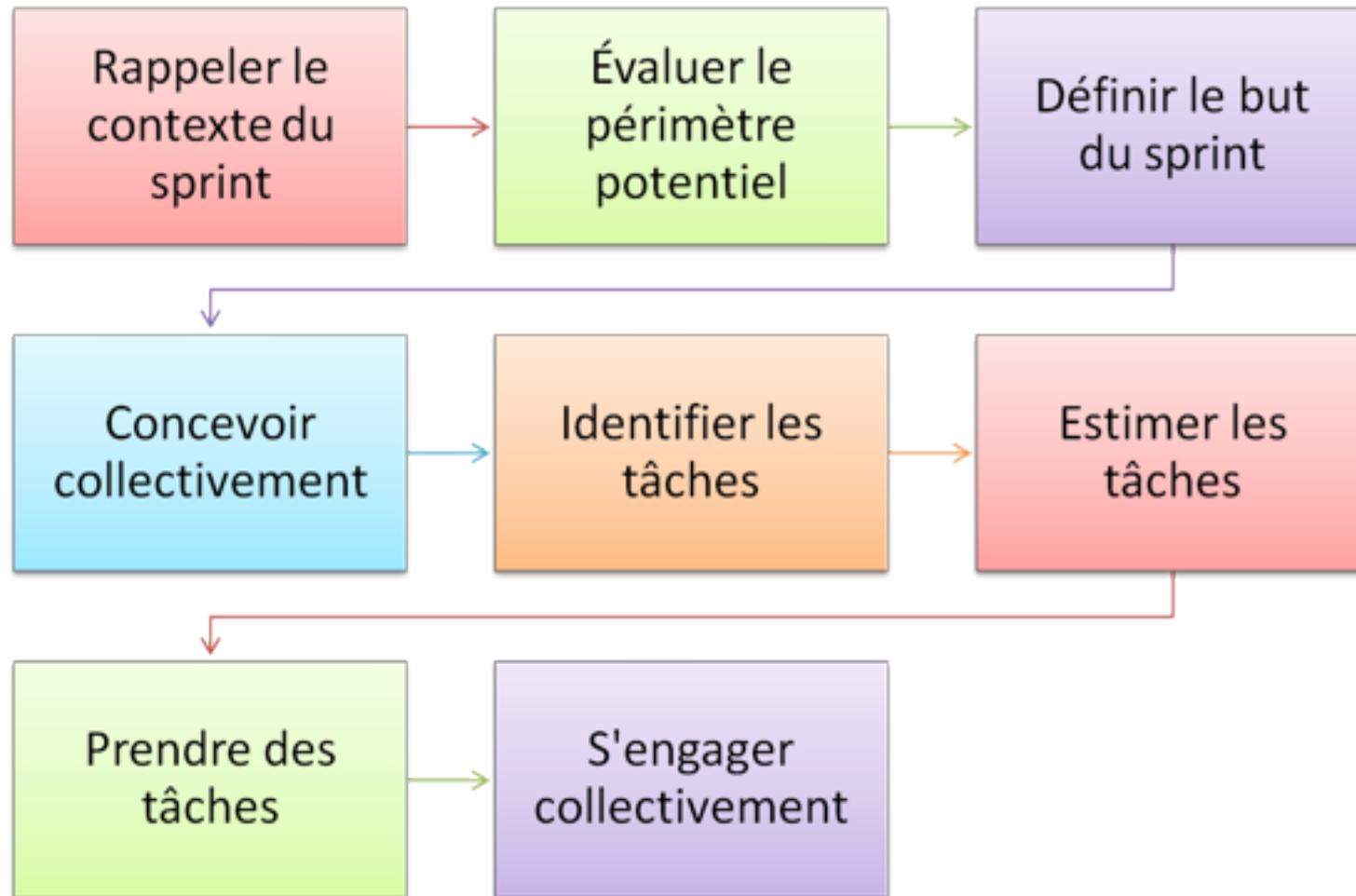


Quand ?

*Réunion de
planification*



Etapes



Contexte du sprint

Sprint3

Début 2 mars - Fin le 15 mars

Disponibilité de l'équipe :

CJ : 10 j

DB : 9j

HG : 10j

AS : 10j

TF : 5j

CB : 10j



Liste des tâches du sprint

backlog

Élément du backlog	Estimation
En tant que responsable d'une tâche, j'estime le reste à faire	2
En tant que responsable d'un projet, je vois l'avancement de toutes les tâches	5
Correction d'un bug sur le démarrage d'une tâche	
En tant que responsable d'un projet, je crée la liste de toutes les tâches	
En tant que responsable d'un projet, je crée un tableau de bord graphique pour suivre le projet	

Liste des tâches

Modifier la page de la tâche	
Ajouter les attributs au modèle métier	
Coder les services	
Ecrire les Junits	
Vérifier les règles métier	
Créer l'arbre dépliant	
Implémenter le calcul de l'avancement	
Implémenter le démarrage d'une tâche prête	

Nouveau à chaque sprint



Tableau des tâches

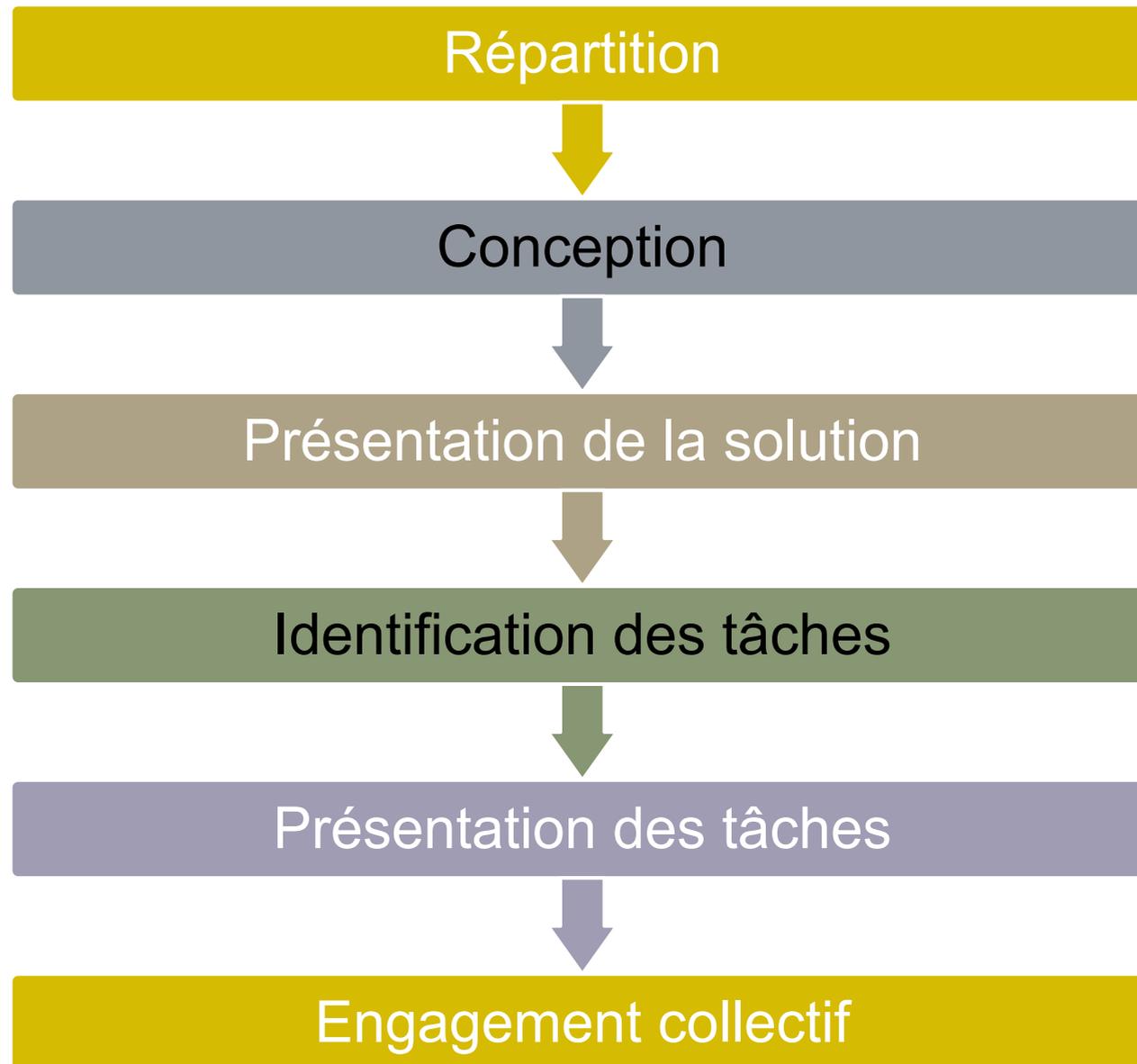
Sprint 3 : début le 15/3, fin le 29/3

But : *lorem ipsum*

	À faire	En cours	Fini
récurrent	tâche tâche		
urgent	tâche		
story	tâche tâche tâche tâche tâche tâche		
story	tâche tâche tâche tâche		
story	tâche tâche tâche tâche tâche		



Conception pendant la réunion



Types de tâches

Associée à une story

- Analyse, conception, codage, test... pour réaliser la story
- Objectif : finir la story

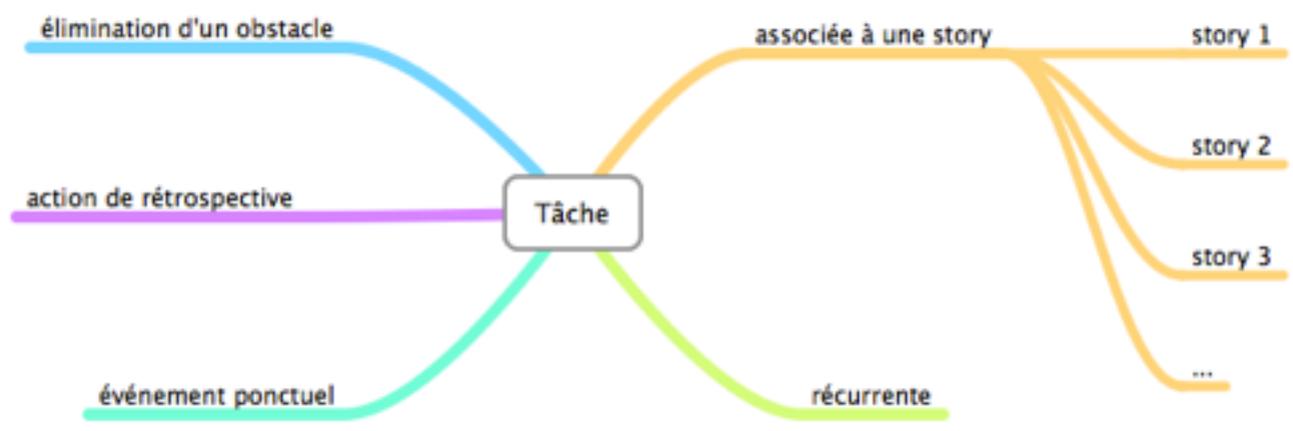
Récurrente

- Revient à chaque sprint
- Objectif : finir le sprint avec le niveau de qualité défini

Urgente

- Non prévue au début du sprint
- Objectif : traiter les urgences qui ne peuvent pas attendre le prochain sprint





Engagement

- ▶ L'équipe s'engage collectivement
 - ▶ à faire de son mieux pour atteindre les objectifs, c'est à dire finir les stories
 - ▶ à obtenir du code de qualité à la fin du sprint
 - ▶ à être transparente sur l'avancement



Conseils

A essayer

- Préparer le backlog en anticipation
- Décomposer en tâches courtes
- Garder du mou
- Faire de la conception

A éviter

- Décider du périmètre à la place de l'équipe
- Ne pas laisser l'équipe identifier les tâches
- Prendre un engagement déraisonnable



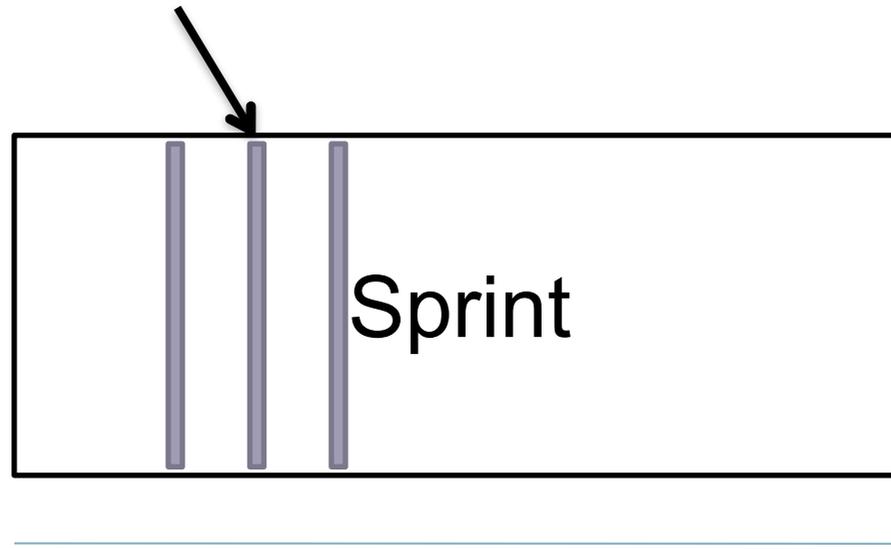
Scrum quotidien

8



Quand ?

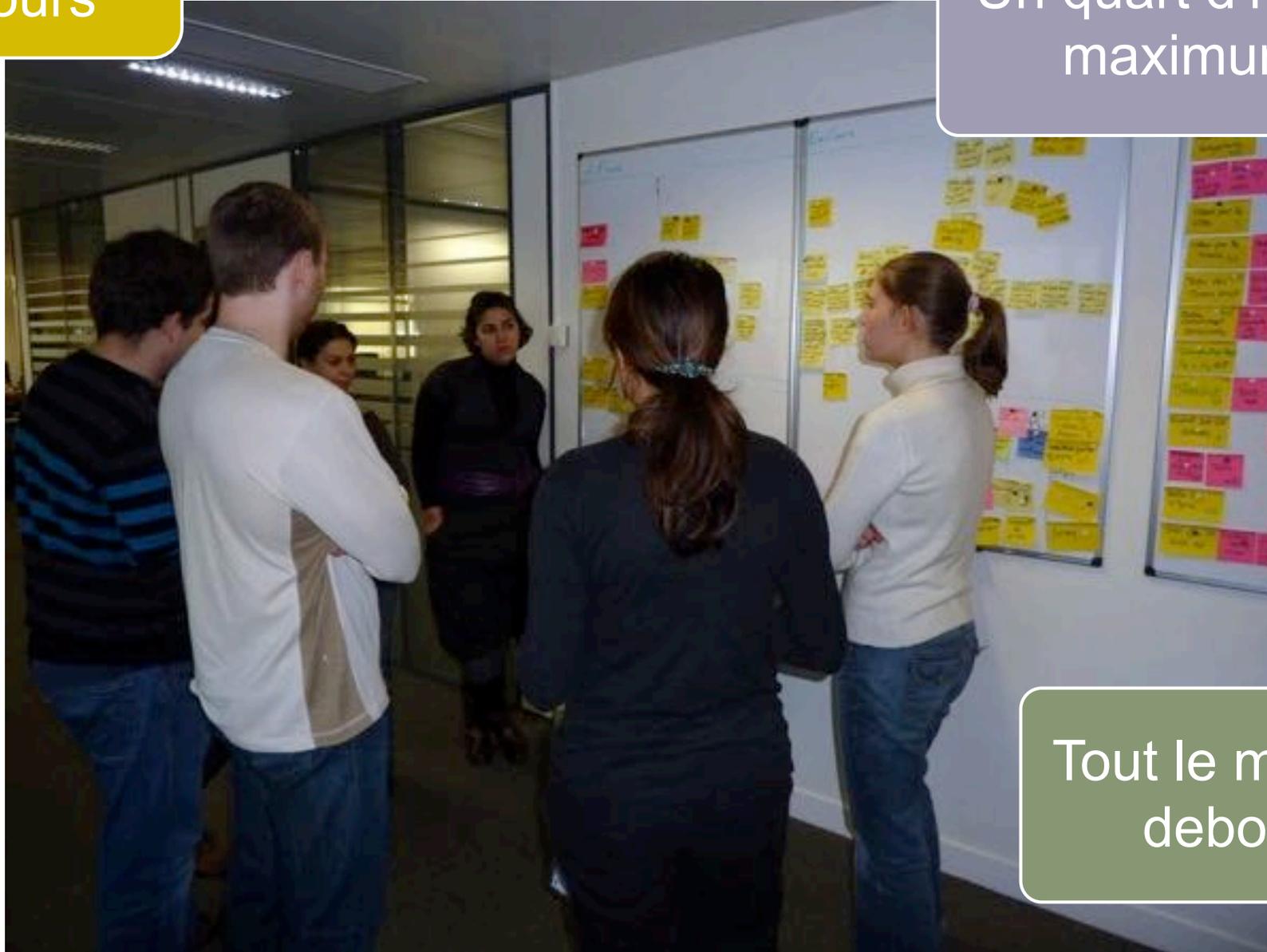
Scrum
quotidien



Scrum quotidien

Tous les jours

Un quart d'heure maximum



Tout le monde debout



Chacun s'exprime sur

Les travaux effectués

Les travaux prévus

Les obstacles rencontrés



Suivi en heures

Tâches de la story S	Lun	Mar	Mer	Jeu	Ven
Coder l'IHM	8	4	8		
Coder couche métier	16	12	10	4	
Tester l'intégration	8	16	16	11	8
Ecrire l'aide en ligne	12				
Coder couche service	8	8	8	8	8
Tracer les erreurs			8	4	



Burndown chart de sprint

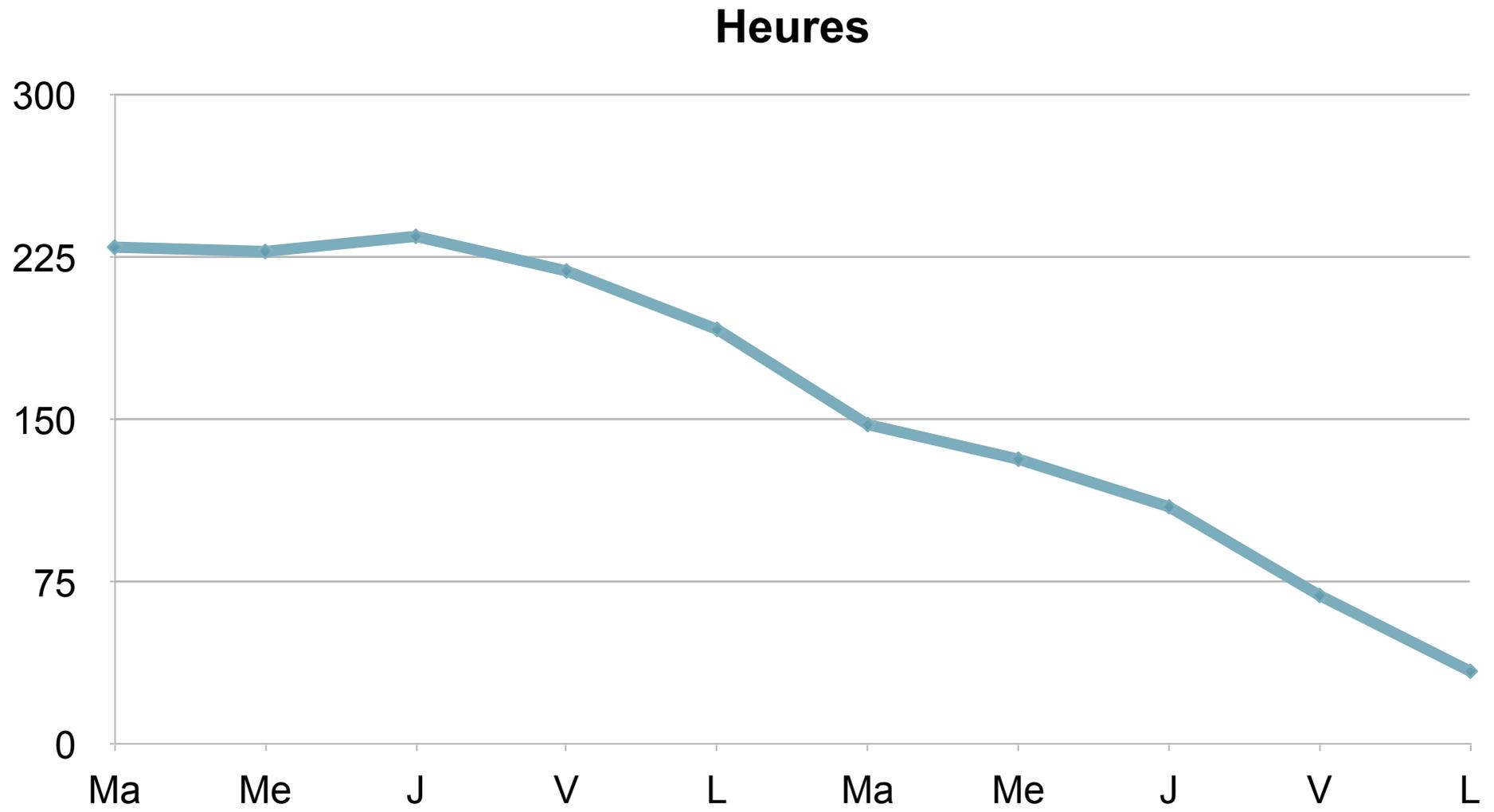
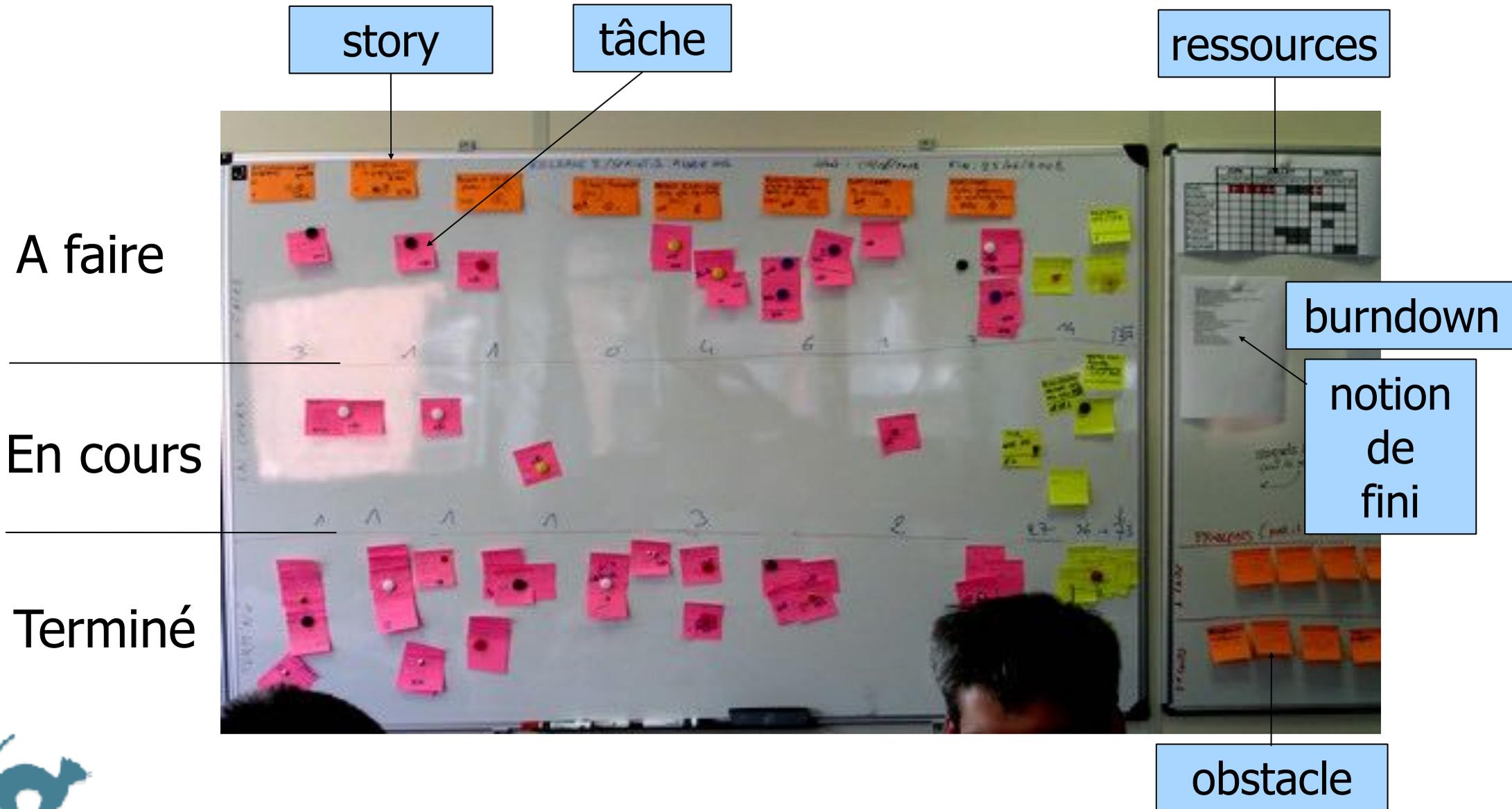


Tableau des tâches

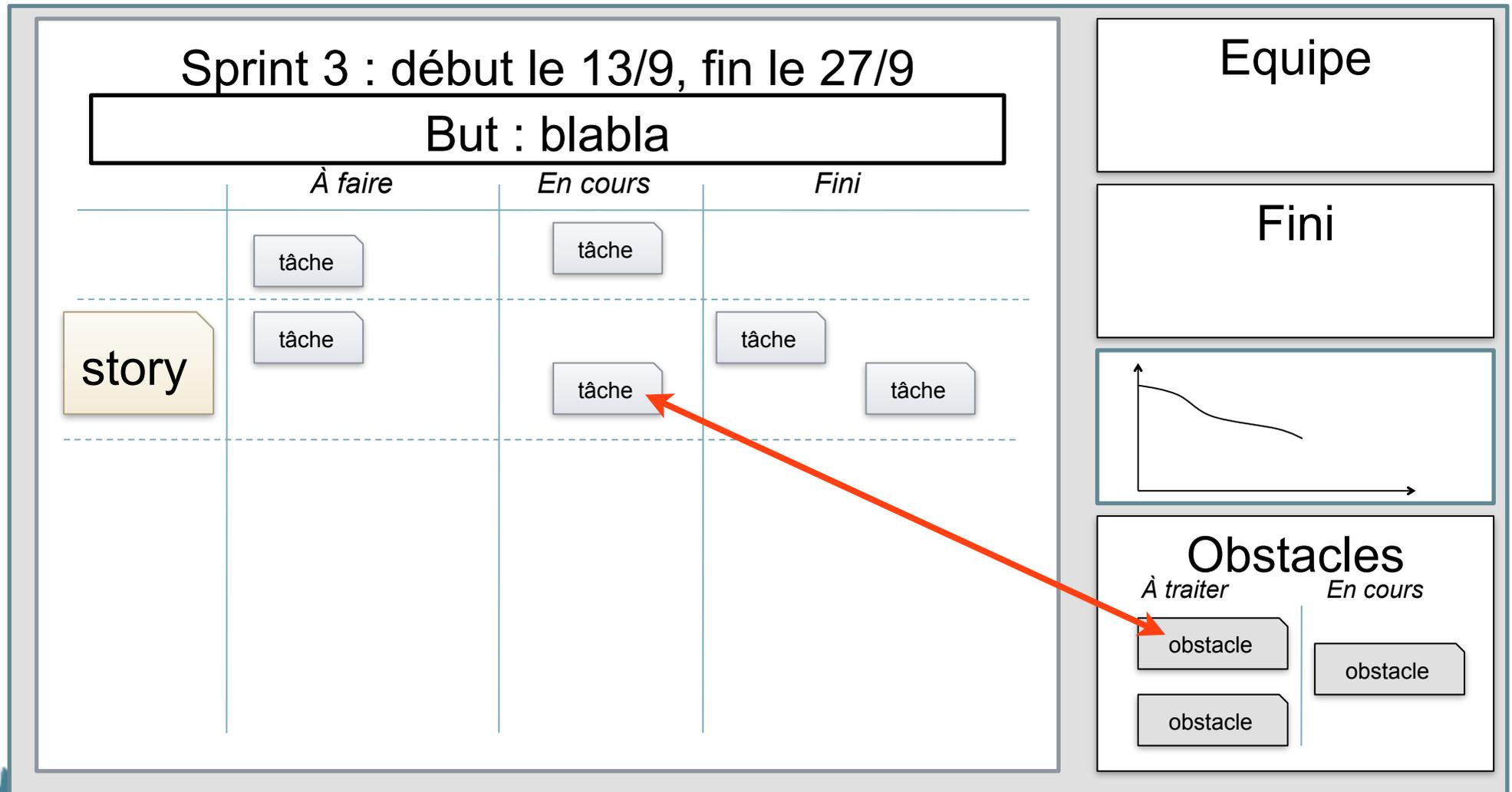


Obstacle

- ▶ Un obstacle est quelque chose qui empêche le bon déroulement du sprint
 - ▶ Il ralentit ou bloque une ou plusieurs tâches
- ▶ Le ScrumMaster a pour mission de faire en sorte que les obstacles soient éliminés



Obstacles



Conseils

A essayer

- Faire le suivi des tâches avec les états
- Organiser des variations dans le déroulement du scrum
- Finir vite les stories

A éviter

- Dépasser un quart d'heure
- S'intéresser au temps passé sur les tâches

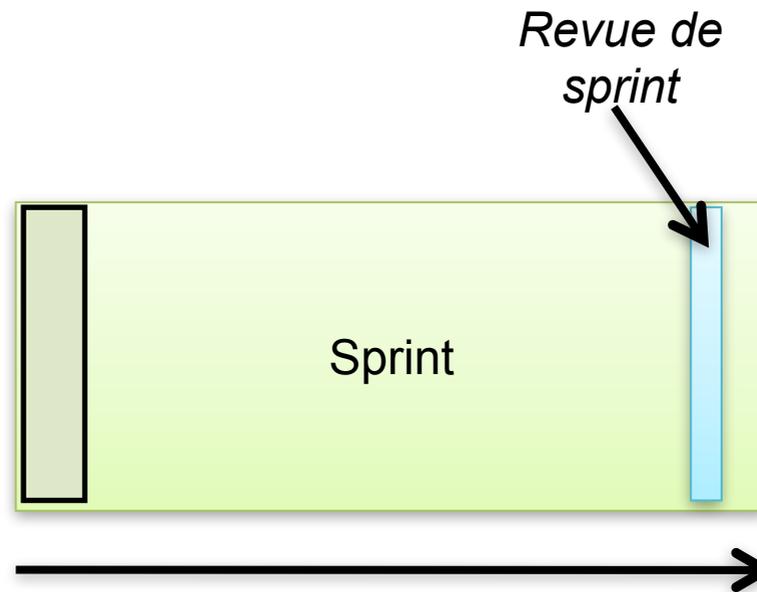


Revue de sprint

9



Place dans le sprint

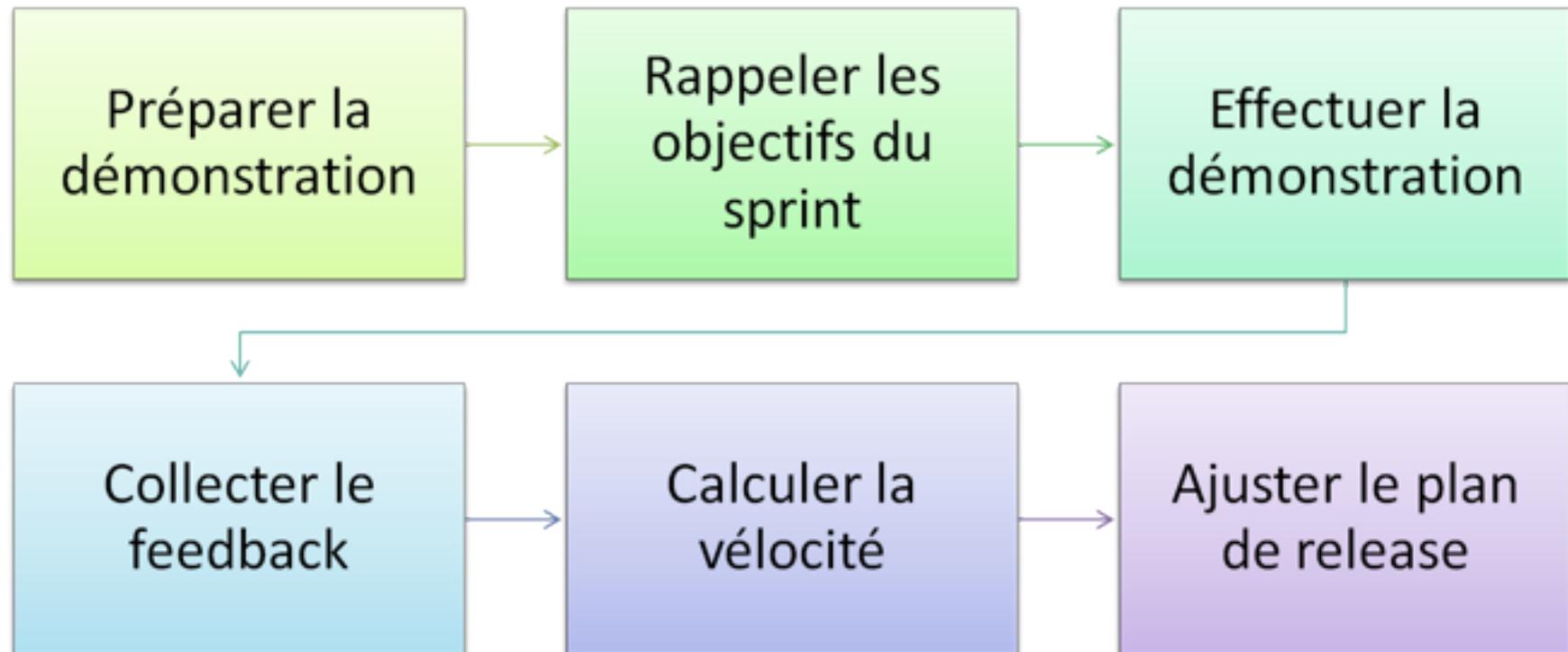


Objectif

Prendre des décisions sur la
poursuite du
développement en
connaissant la situation
actuelle



Etapes



Démonstration

L'équipe présente ce qu'elle a fait pendant le sprint

Se fait avec une démo des nouvelles fonctionnalités

Informel

- Préparation < 2h
- Pas de slides

On invite du monde

Toute l'équipe participe



Vélocité vs capacité

Début du sprint

2

3

5

2

3

3

Fin du sprint

2

3

5

2

3

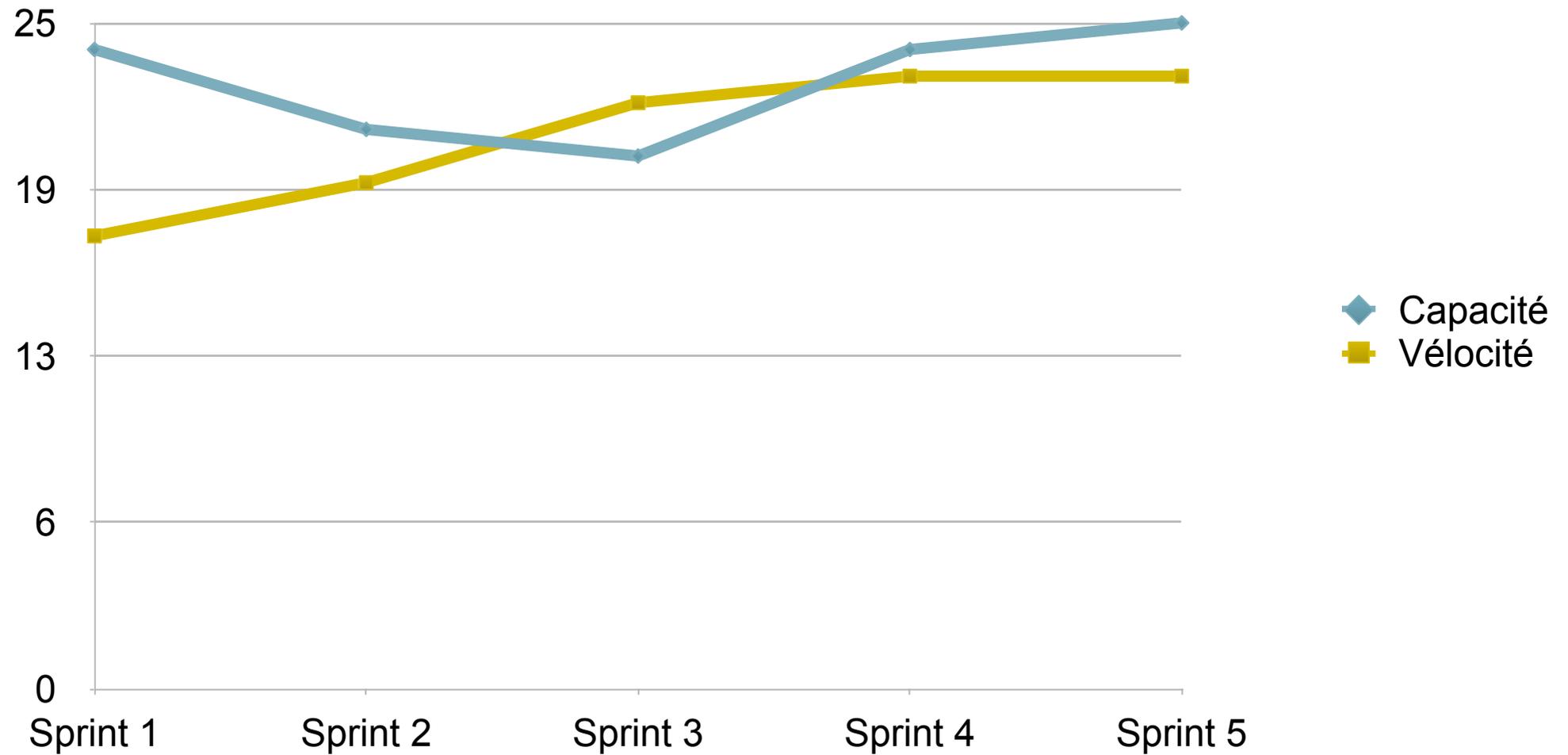
3

Capacité estimée =
18 points

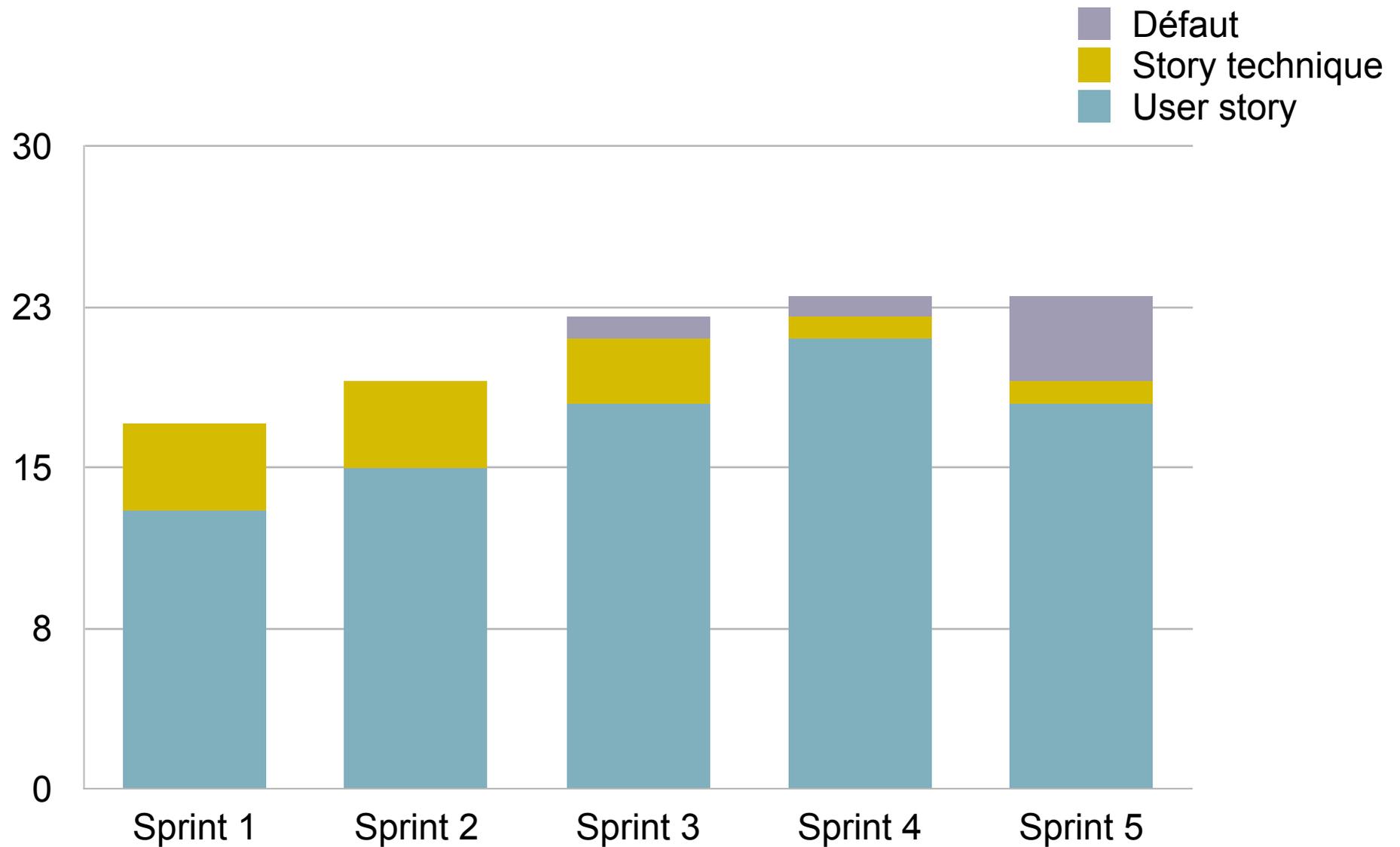
Vélocité réelle =
12 points



Vélocité vs capacité



Vélocité par type de story



Influence sur les plans

Le feed-back a pour objectif d'améliorer le produit



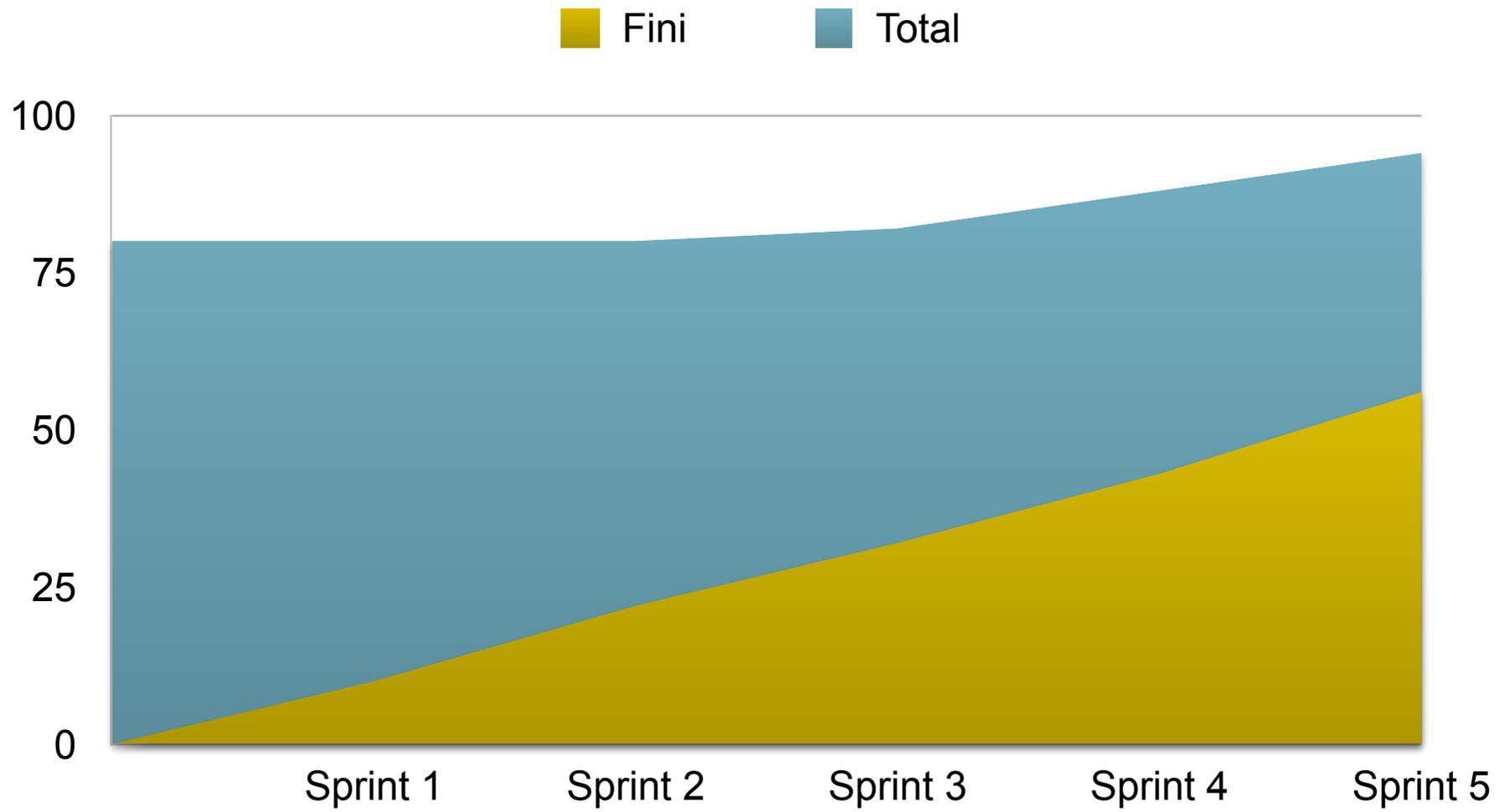
Les améliorations passent par des ajouts dans le backlog



La taille du backlog va augmenter...



Burnup



Conseils

A essayer

- Dérouler un scénario
- Inviter largement mais expliquer que c'est un produit partiel
- Parler des stories
- Solliciter le feedback
- En faire une réunion essentielle sur le produit

A éviter

- Modifier le produit au dernier moment
- Parler processus
- Parler des tâches
- Ne pas finir les stories
- Faire un compte-rendu



Rétrospective

10

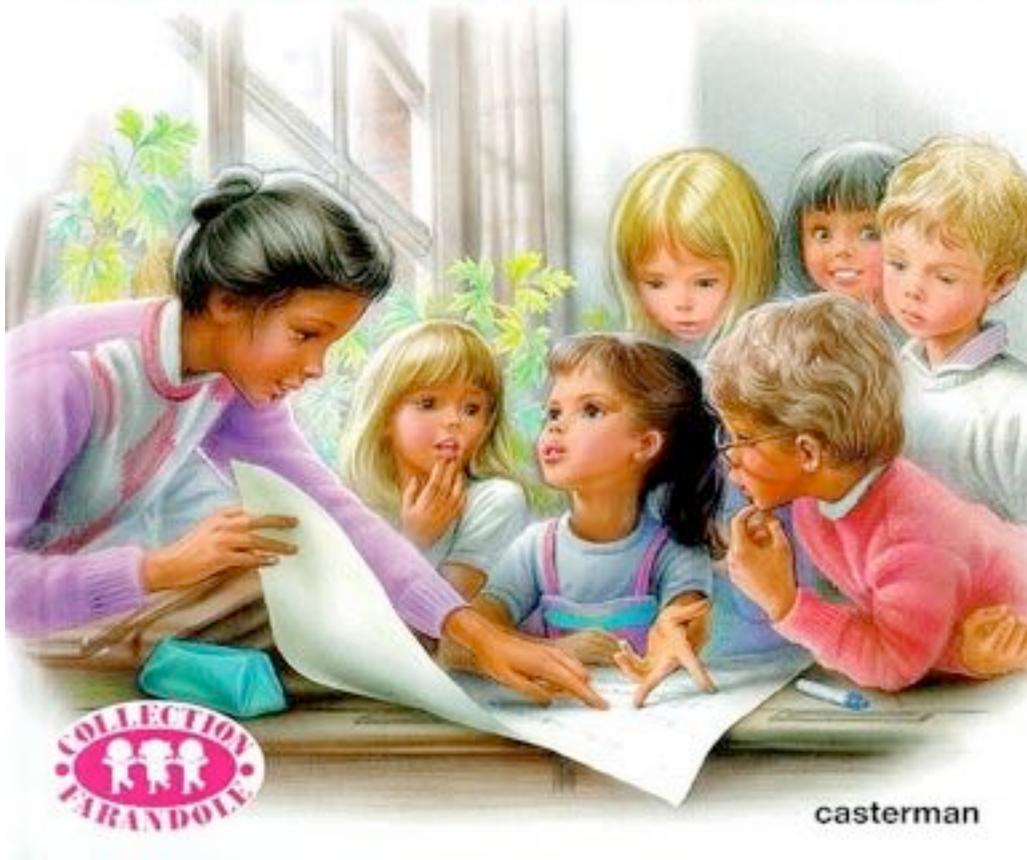


Objectif

GILBERT DELAHAYE - MARCEL MARLIER

martine

Fait la retrospective de son projet Scrum



A intervalles réguliers, l'équipe réfléchit à comment devenir plus efficace, puis adapte et ajuste son comportement en conséquence



Exemple de questions

Commencer à faire

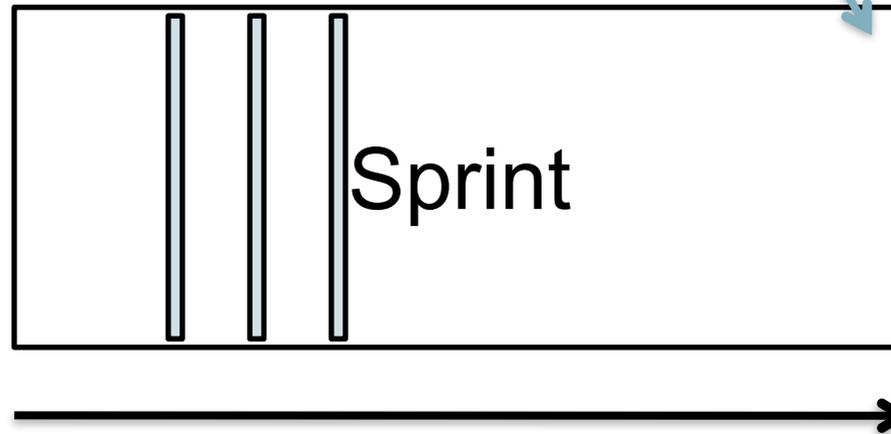
Arrêter de faire

Continuer à faire

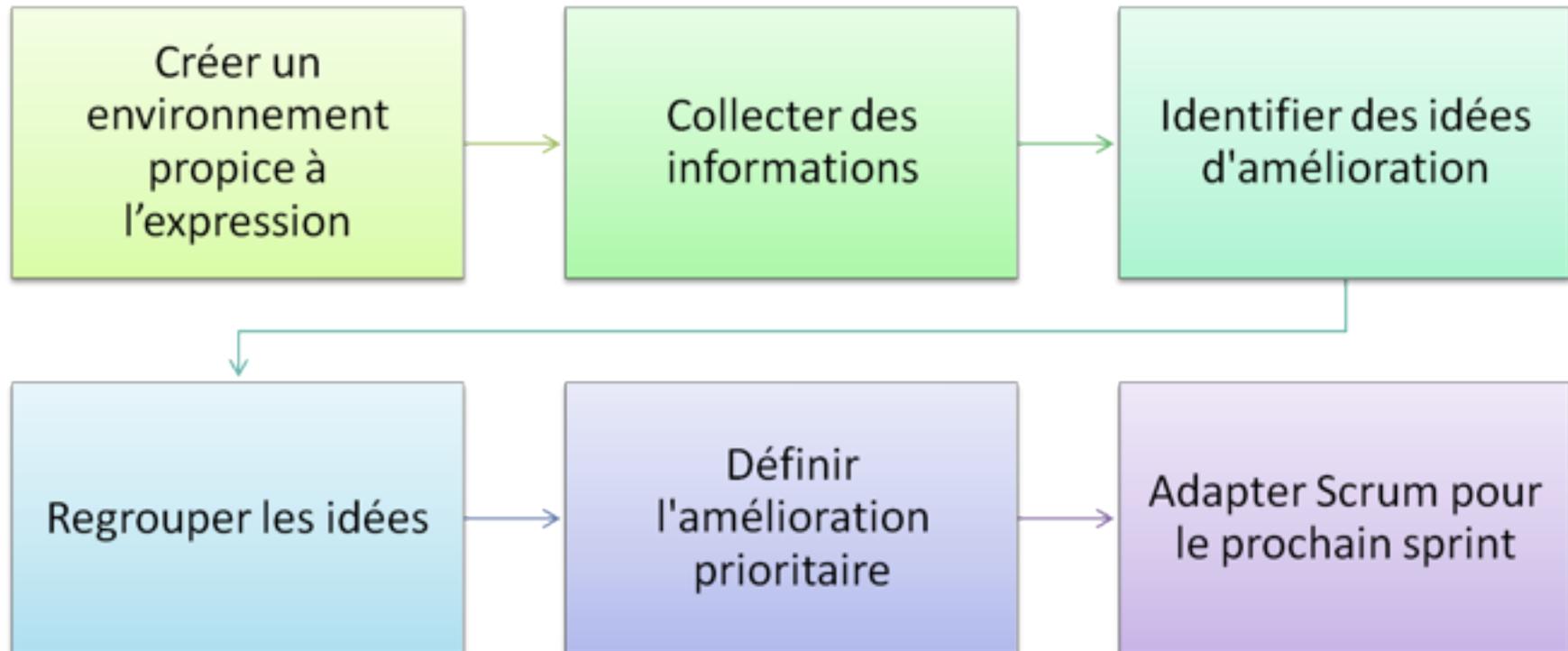


Quand ?

Rétrospective



Etapes



Conseils

A essayer

- Parler de ce qui va bien
- Se concentrer sur une seule amélioration
- Mener des rétrospectives plus poussées en fin de release
- Utiliser un facilitateur externe

A éviter

- Régler ses comptes
- Vouloir tout améliorer d'un coup
- Ne pas faire aboutir les actions décidées
- Arrêter de faire des rétrospectives

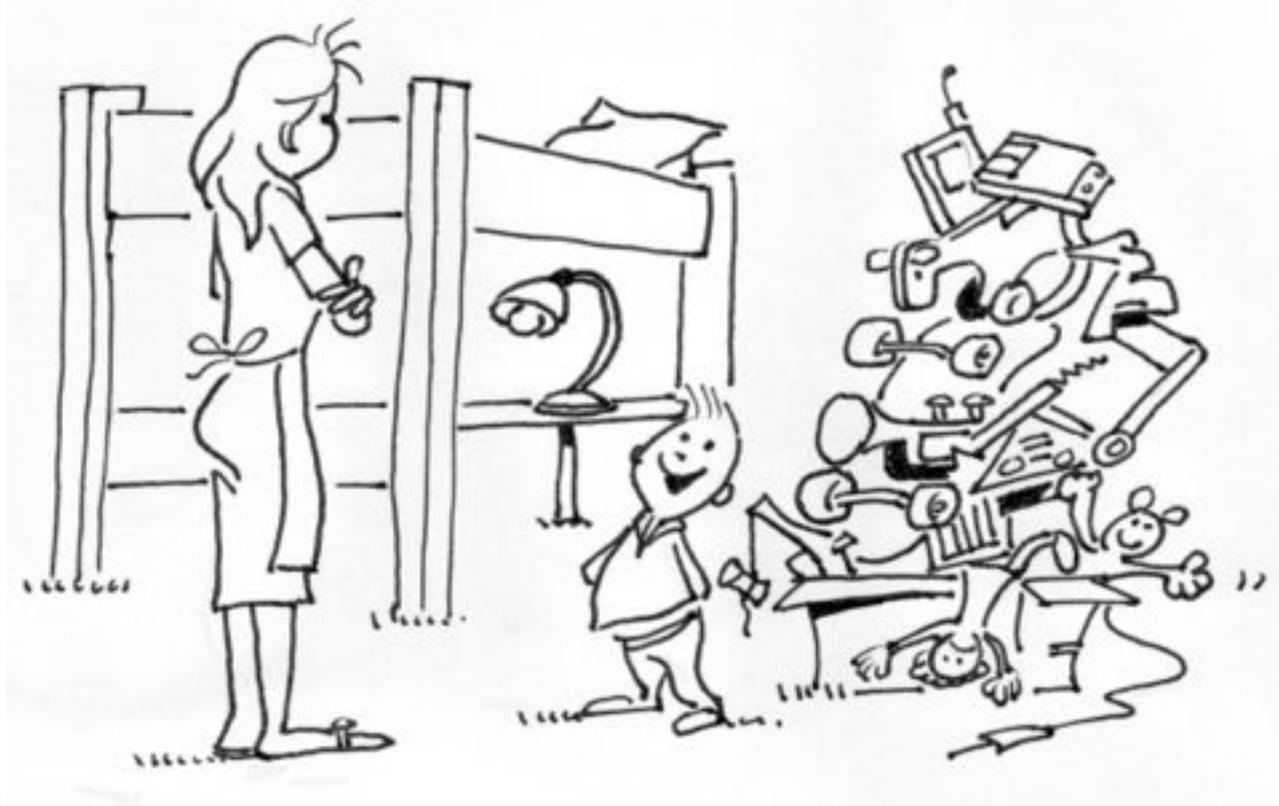


Signification de fini

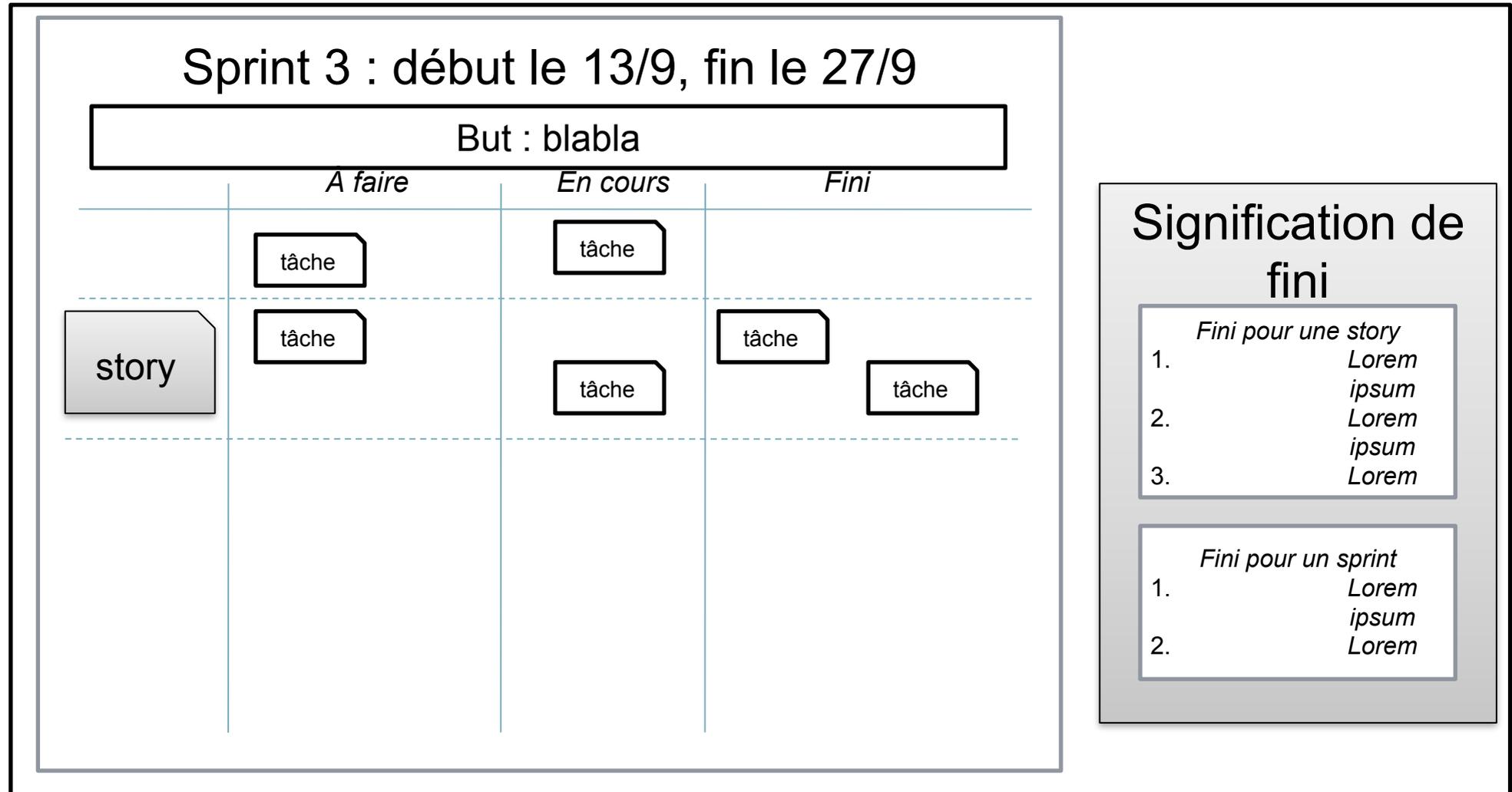
11



Fini ?



Doit être visible



Signification de fini

Fini pour une *story*

- Les tests passés avec succès
- Les autres contrôles relatifs à une *story*

Fini pour un sprint

- Le produit partiel livré dans un environnement de test
- Les autres contrôles

Fini pour une *release*

- Ce qui reste à faire pour la mise en production



Des stories aux tests d'acceptation 14



Réduire le délai

développement

test

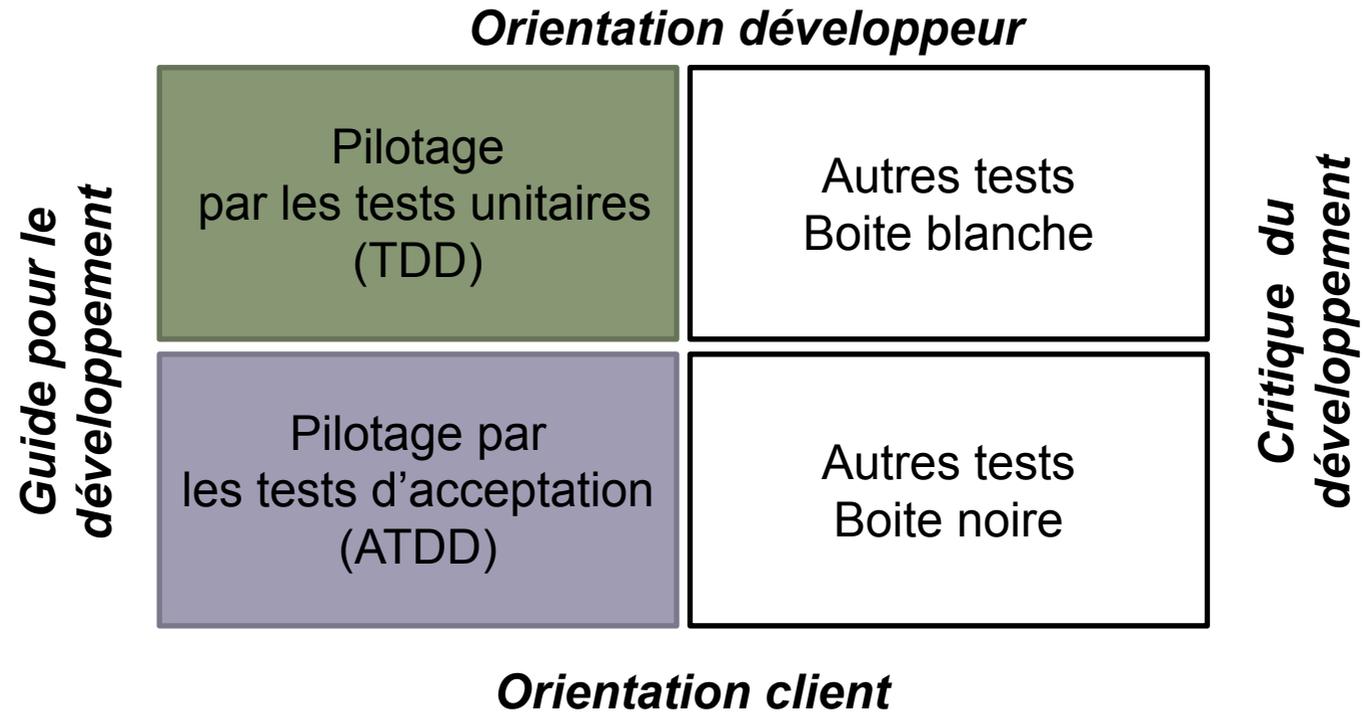
correction



développement test correction



Quadrant des tests



D'après Brian Marick



2 catégories de tests

acceptation

Tests associés à une exigence story qui guident son développement de bout en bout et permettent de la considérer comme finie

unitaire

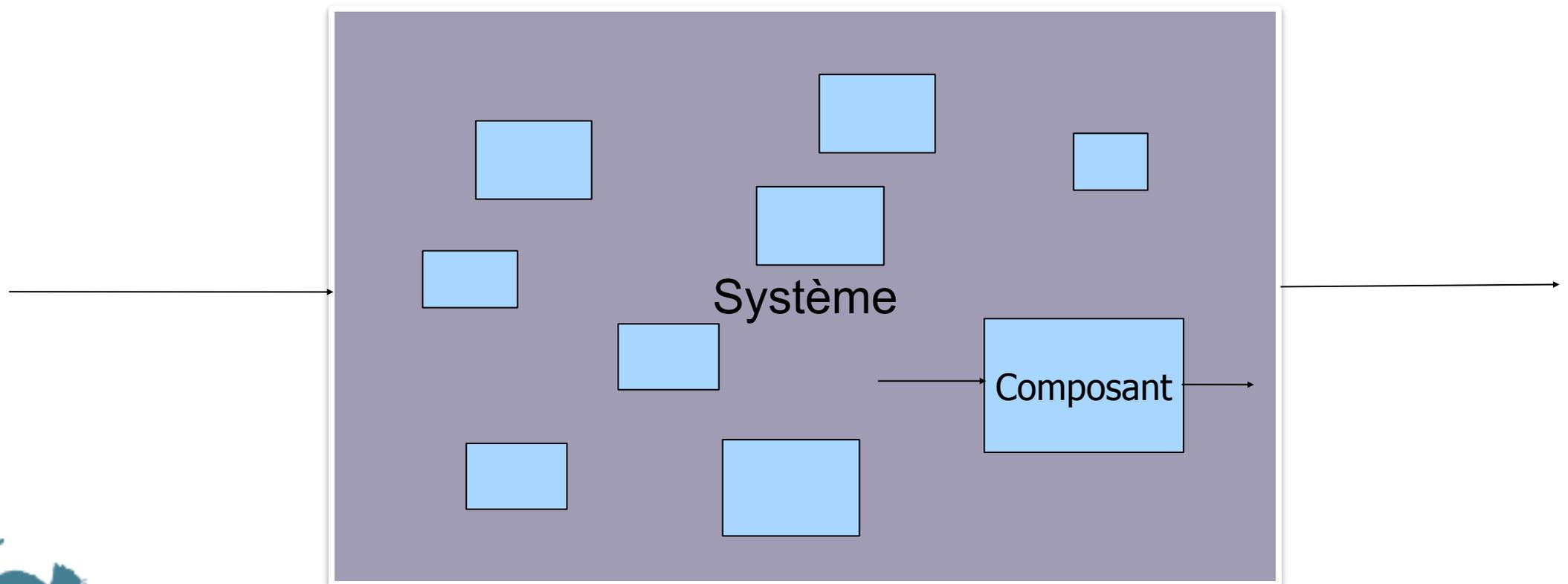
Tests qui guident le développeur pendant la conception et le codage



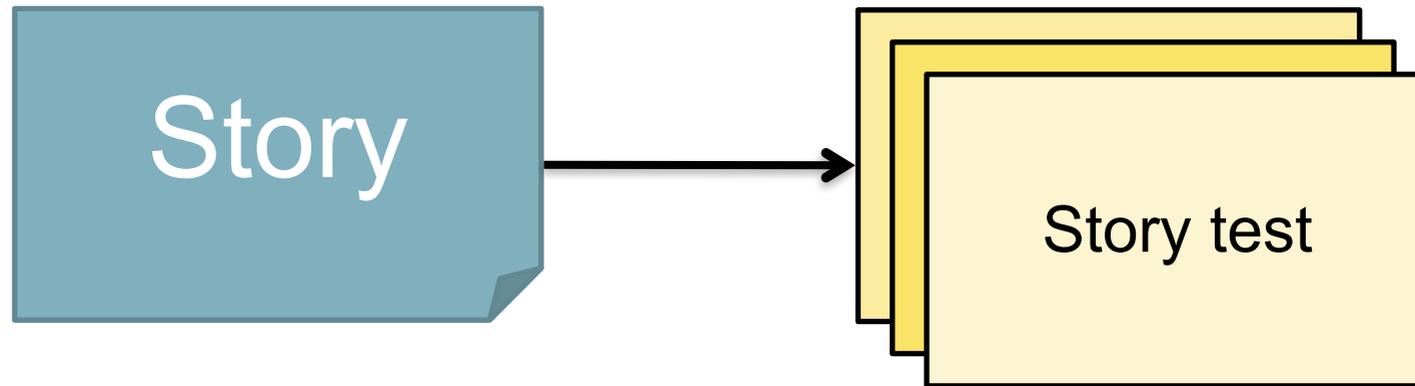
2 niveaux d'application

acceptation

unitaire



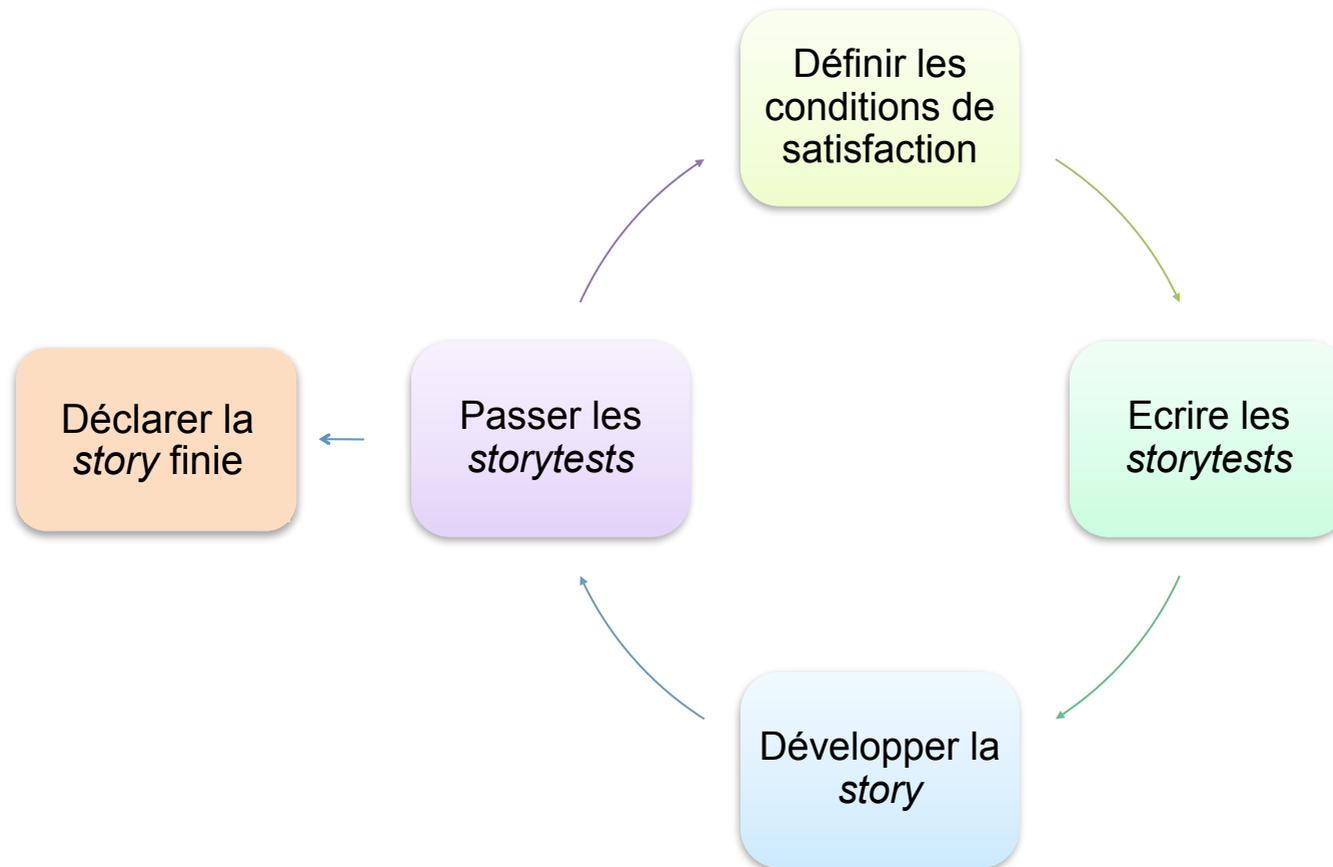
A(cceptance)TDD ?



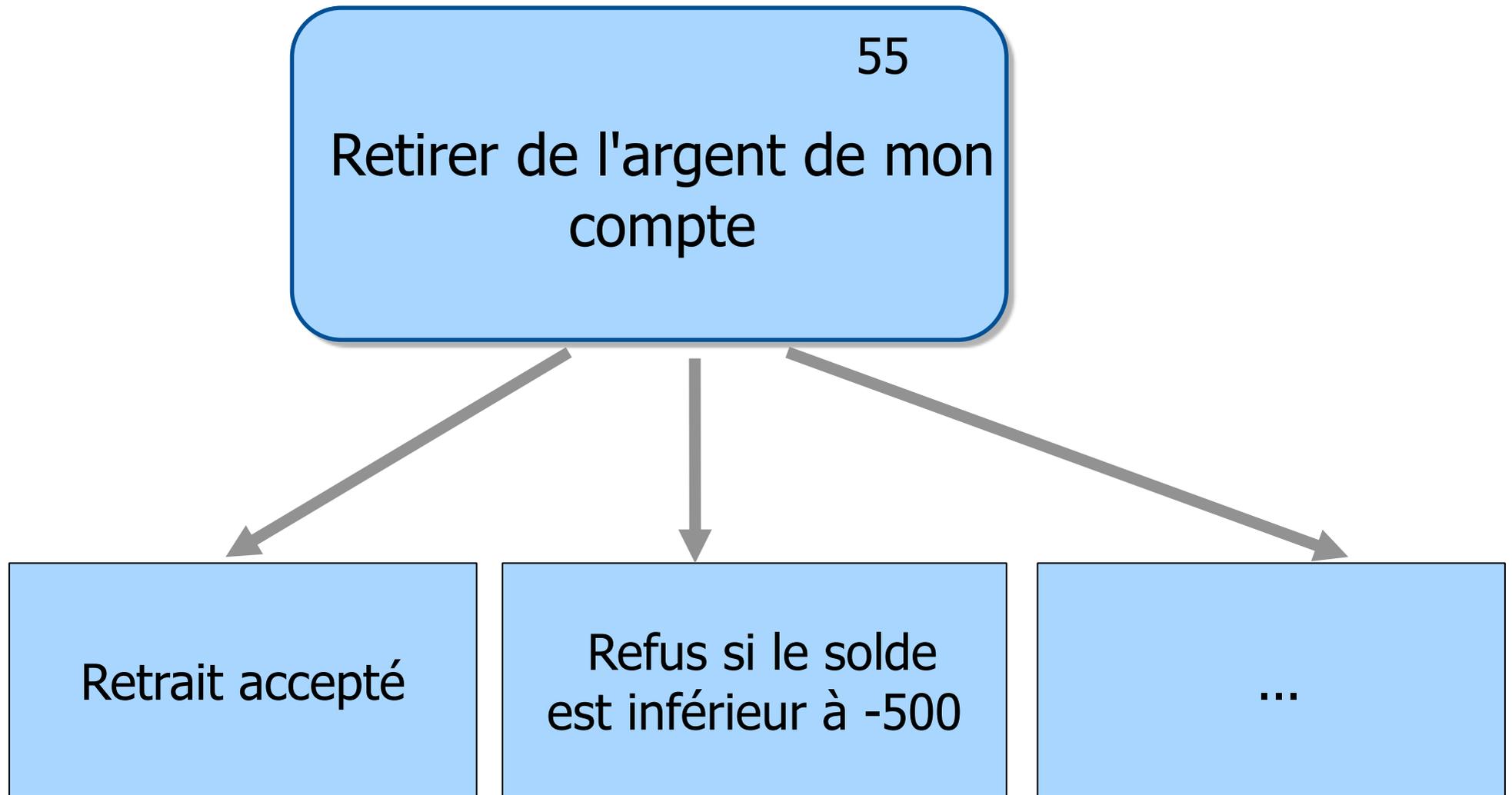
Traçabilidade



Démarche



Conditions de satisfaction



Formaliser les conditions

Behavior Driven Development

StoryTest x

Etant donné [le contexte]
et [la suite du contexte]

Quand [l'événement]

Alors [résultat]
et [autre résultat]



StoryTest en BDD

Refus si le solde
est inférieur à -500

Etant donné un compte 67890 avec un solde initial de 0 et un client C possesseur du compte

Quand le client C fait un retrait de 600 sur le compte 67890

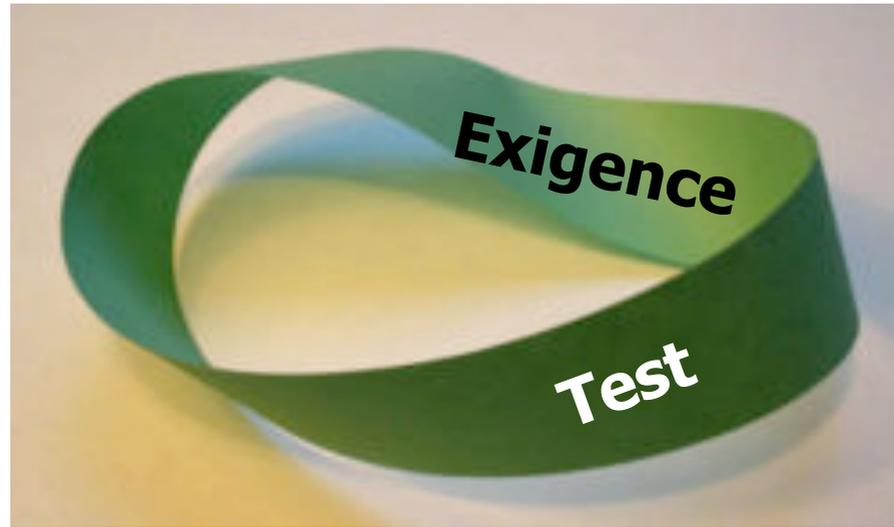
Alors le retrait est refusé

et le message « Vous n'êtes pas autorisé à retirer cette somme » est envoyé à C

et le solde de 67890 reste à 0



Equivalence



Intérêt de l'automatisation

- ▶ Itération 1 : T_1
- ▶ Itération 2 : $T_2 + T_1$
- ▶ Itération 3 : $T_3 + T_2 + T_1$
- ▶ Itération n : $T_n + \dots + T_3 + T_2 + T_1$

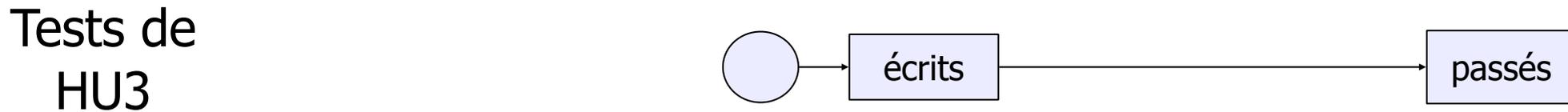
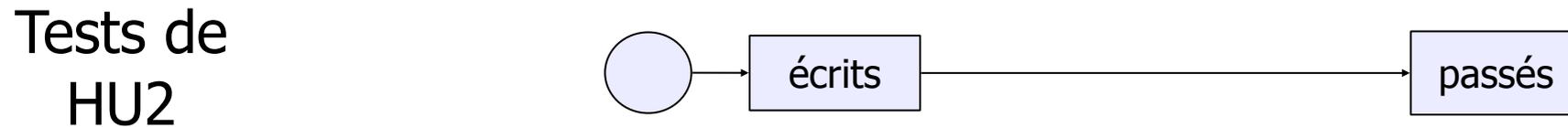
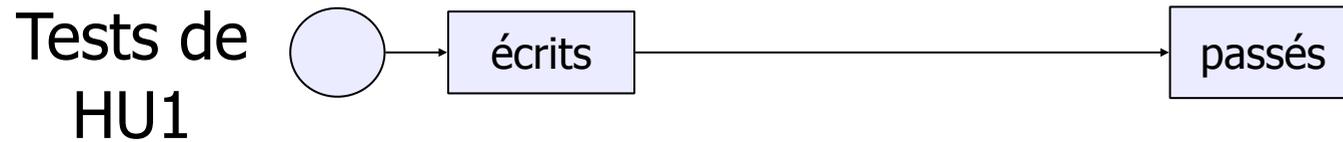
$$\text{Total} = T_i * n * (n+1) / 2$$

$$T_i = 10, n = 10, \text{Total} = 550$$

$$T_i = 50, n = 10, \text{Total} = 2750$$



Processus de test



Principes

Prêts avant
que le codage
démontre

- Ils servent de
spécification



Tous
disponibles au
milieu de
l'itération

- Sinon les
programmeur
s aident à les
écrire

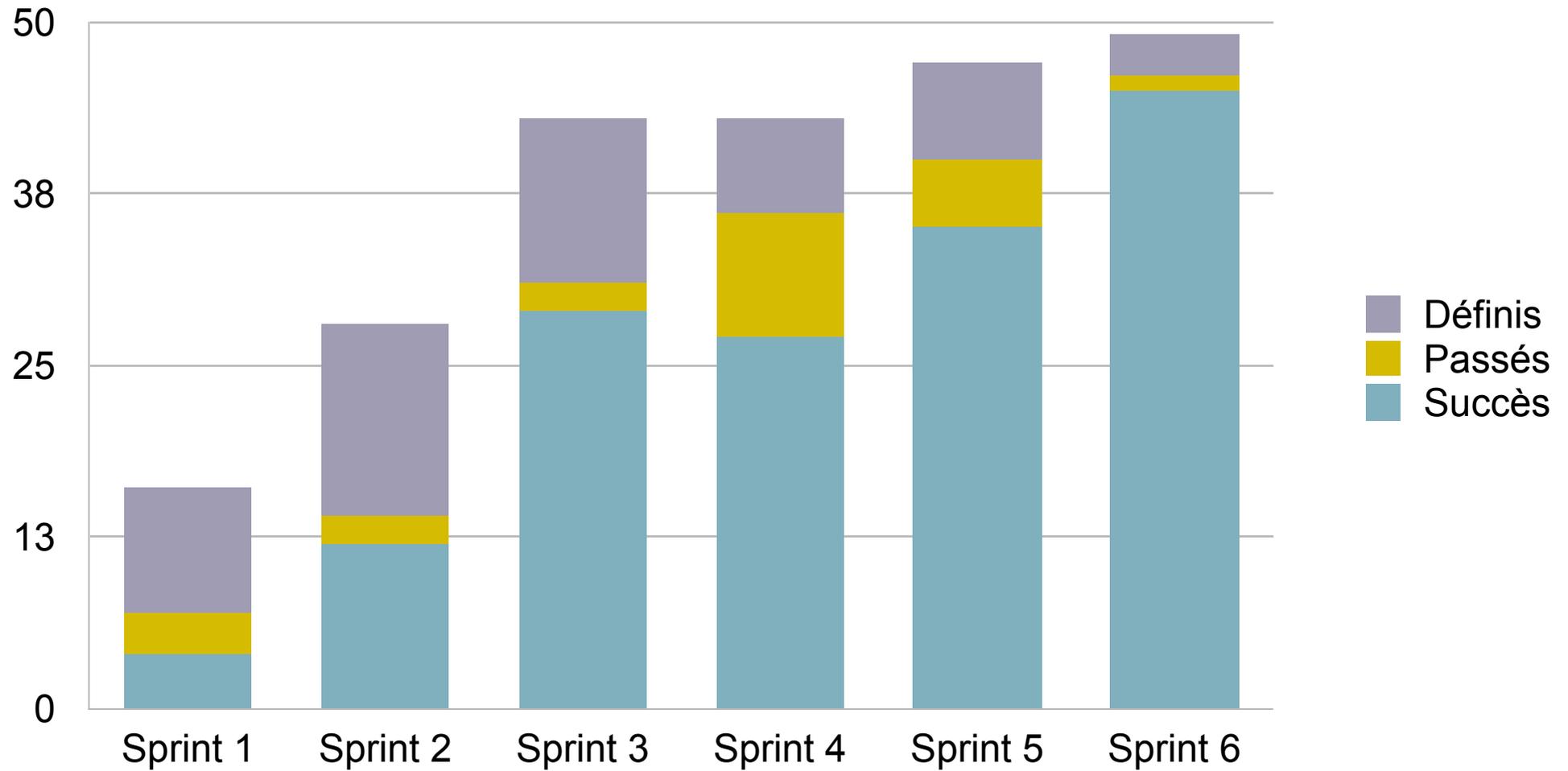


Tous passent
avec succès à
la fin de
l'itération

- Sinon la story
n'est pas finie



Indicateur de suivi des tests



Principes du test agile

Réduire la boucle de feed-back

Le test fait avancer le projet

Le test, ce n'est pas une phase

Tout le monde teste, pas seulement les testeurs

Les tests représentent des attentes

On ne stocke pas les bugs

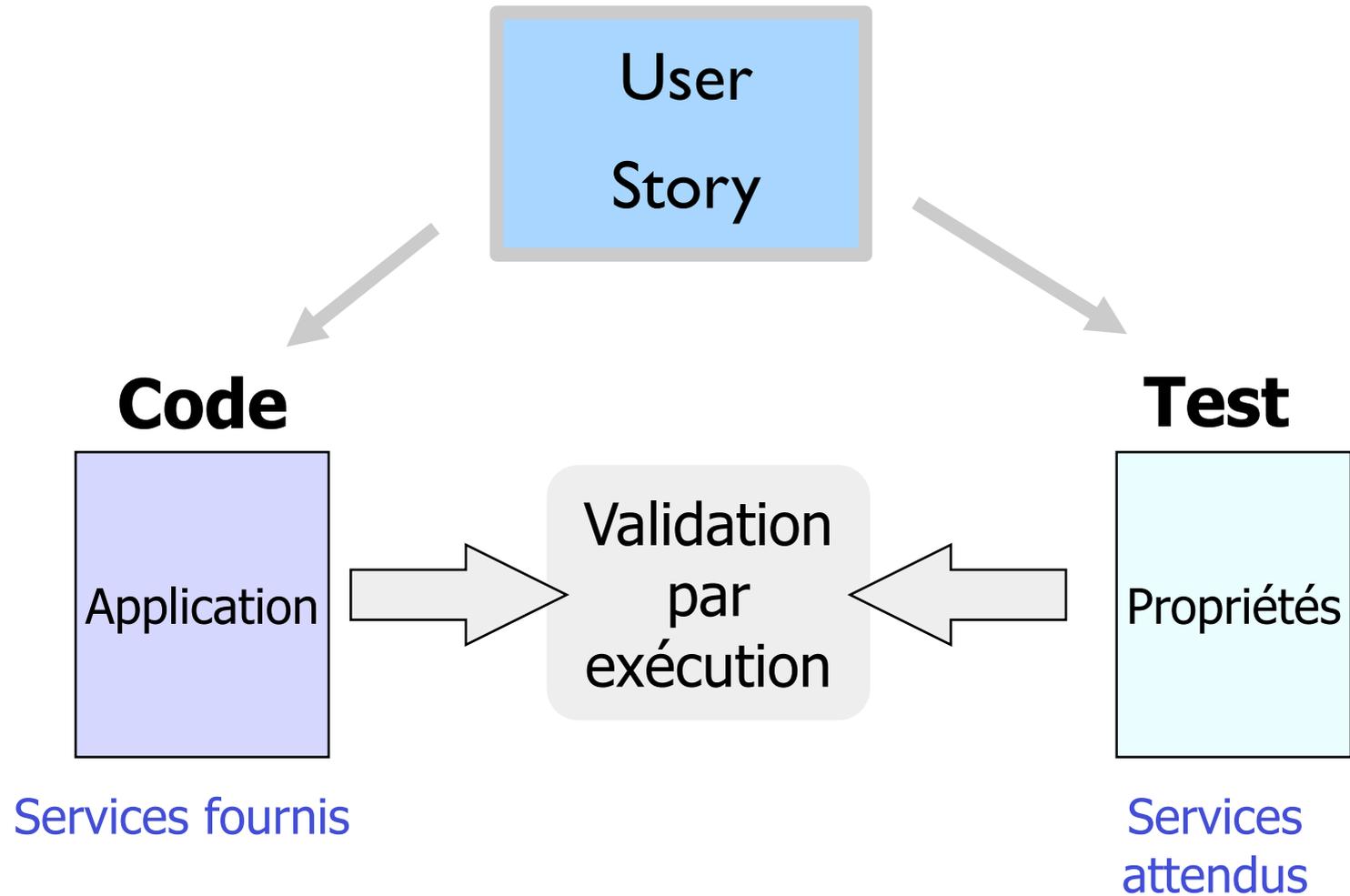
Réduire l'excès de documentation de test

Tester est inclus dans la signification de fini

Définir les tests avant plutôt qu'après



Test fonctionnel automatisé

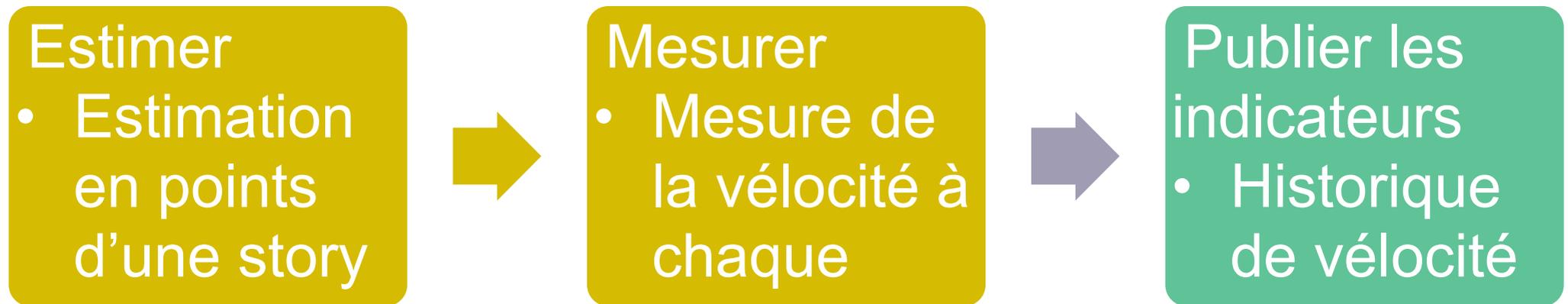


Estimations, mesures et indicateurs

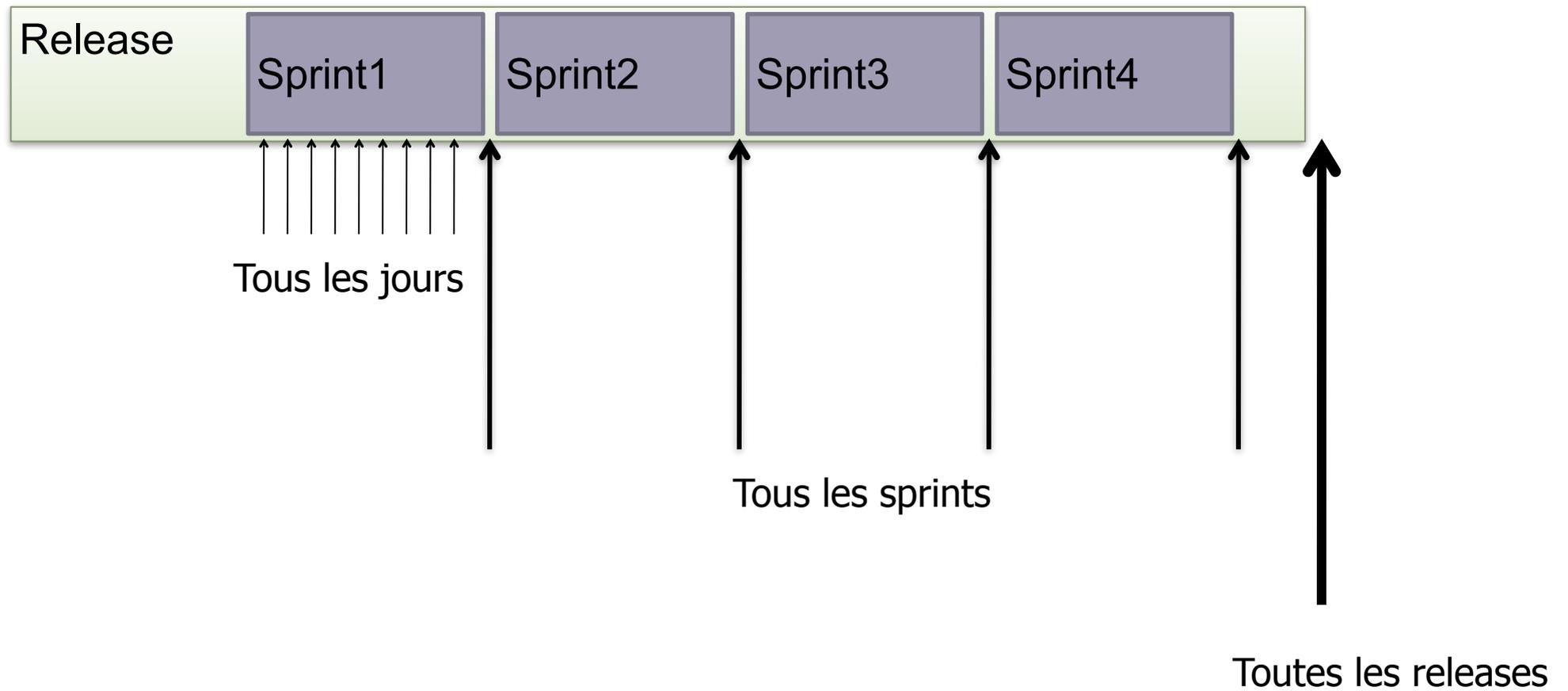
15



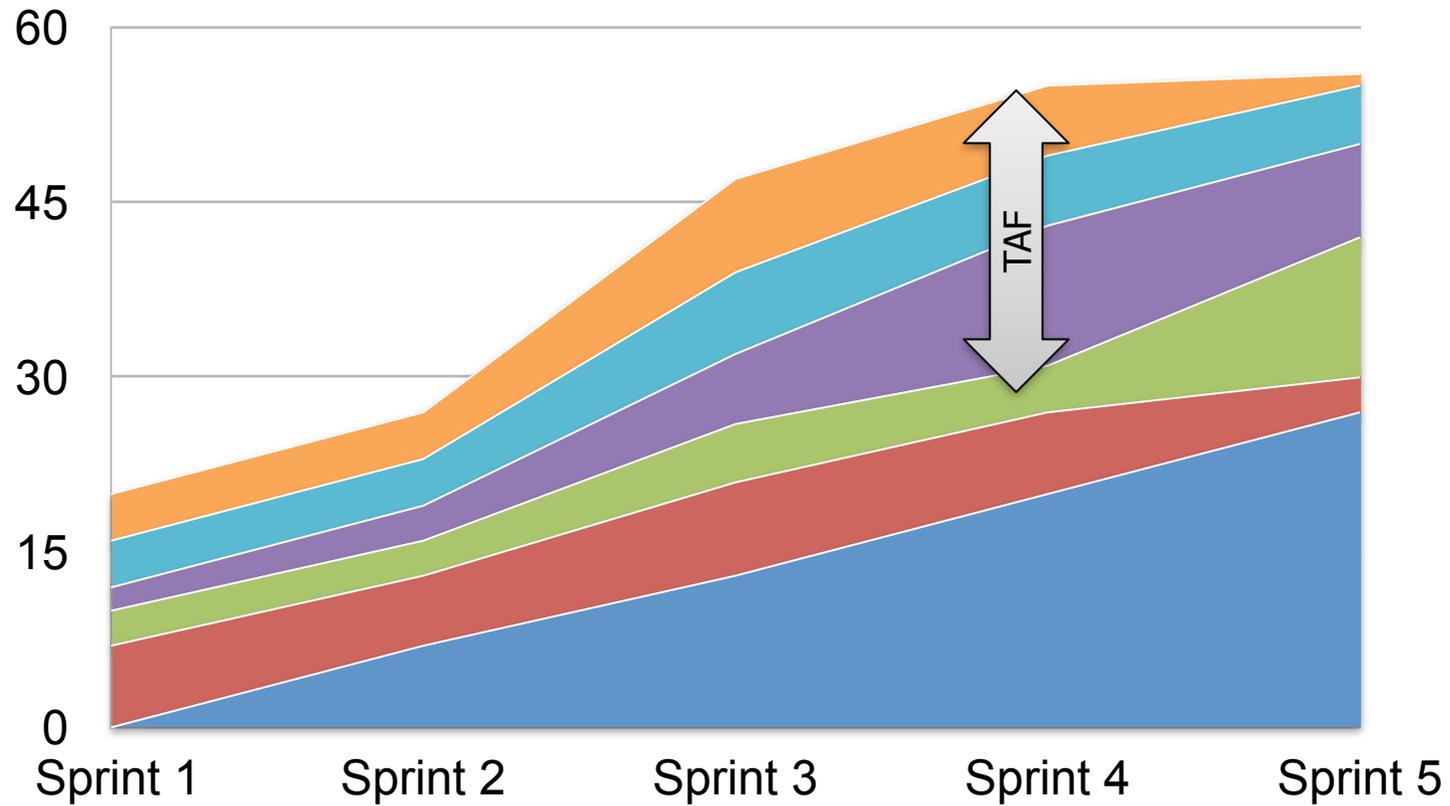
Estimation, mesure, indicateur



Mesures



Flux



- Fini
- En cours
- Planifié
- Estimé
- Accepté
- Identifié



Ingénierie du logiciel 16



Pas de doc, pas de modèle ?

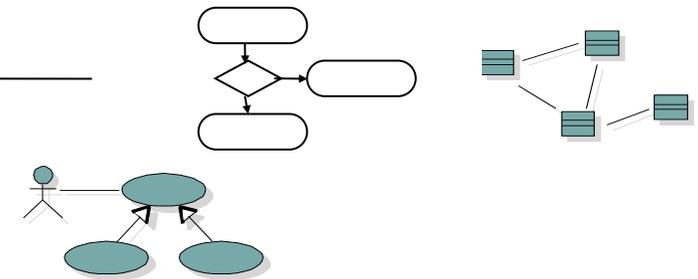
- ▶ Attitude extrême
 - ▶ Pas de doc !
 - ▶ Pas de modèle !

- ▶ Modélisation agile
 - ▶ Utilisation partielle d'UML
 - ▶ Recherche de la communication plutôt que de l'exhaustivité

Expression des exigences

Modèle métier

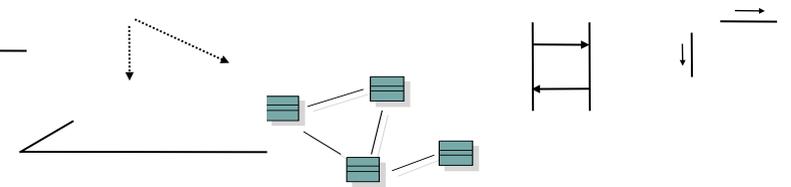
Modèle de cas



Analyse et conception

Modèle d'analyse

Modèle de conception



Modélisation agile

Faire participer l'équipe à la modélisation

Communiquer les modèles

Utiliser le bon diagramme

Mettre l'information à un seul endroit

Utiliser les outils les plus simples

“Prouver” le modèle avec du code



Documentation technique

La Vision

Le code et
les xUnits

Le document
d'architecture

Les stories
avec leurs
tests



Intérêt des tests unitaires



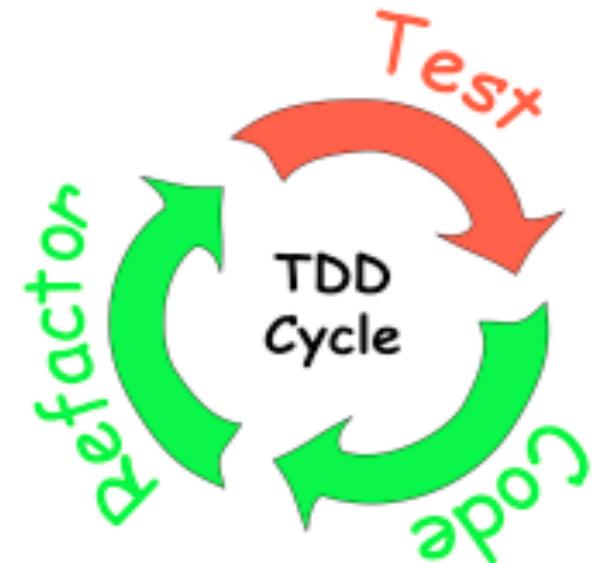
Ils sont passés très vite



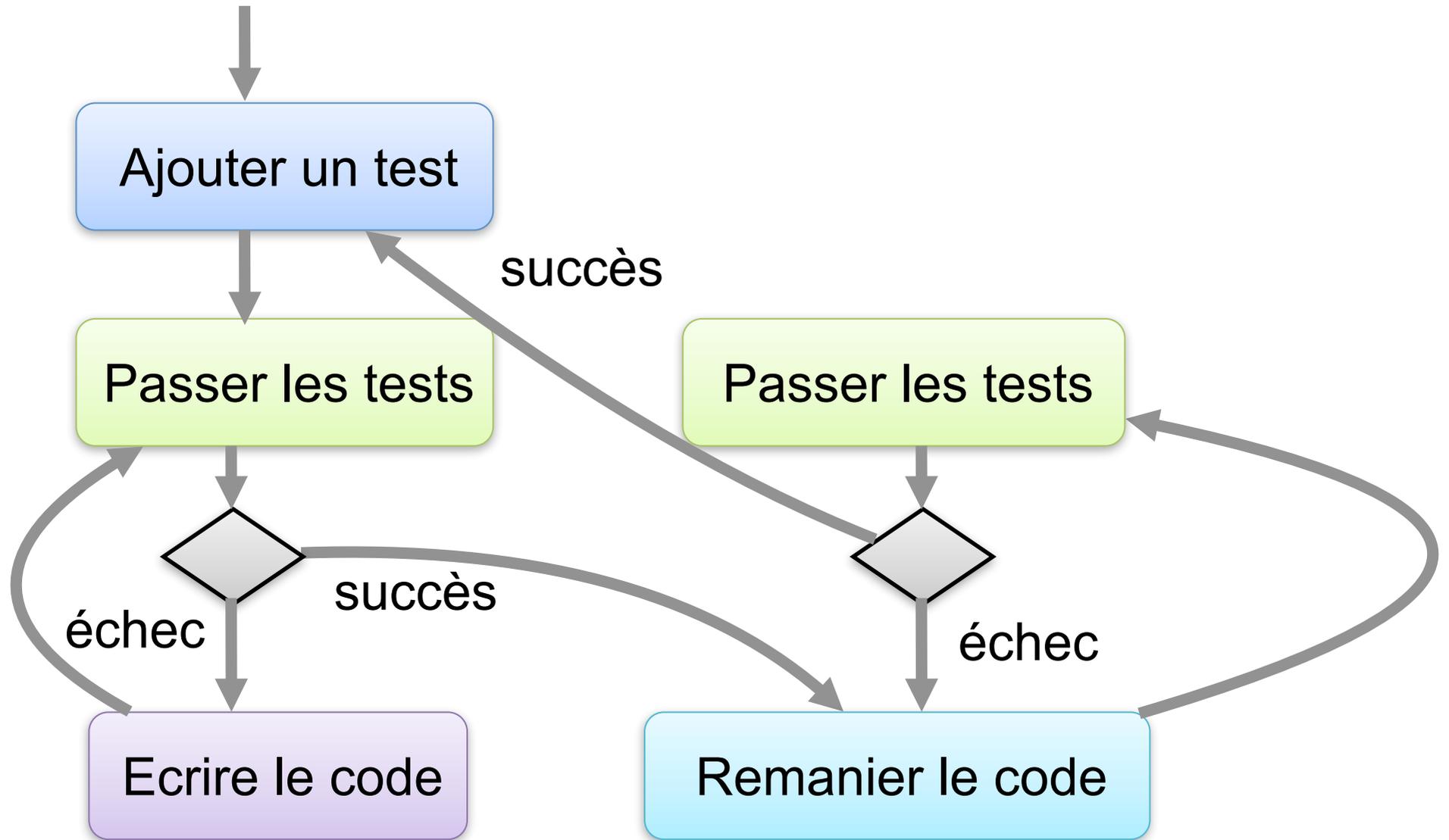
Ils aident à localiser les problèmes



- ▶ Développement piloté par les tests
 - ▶ Une démarche qui va plus loin que l'activité de test
- ▶ La voie
 - ▶ Écrire le test
 - ▶ Penser, inventer, se planter, modifier
 - ▶ Écrire le code pour qu'il passe
 - ▶ Rester simple
 - ▶ Améliorer la qualité
 - ▶ Remaniement



TDD

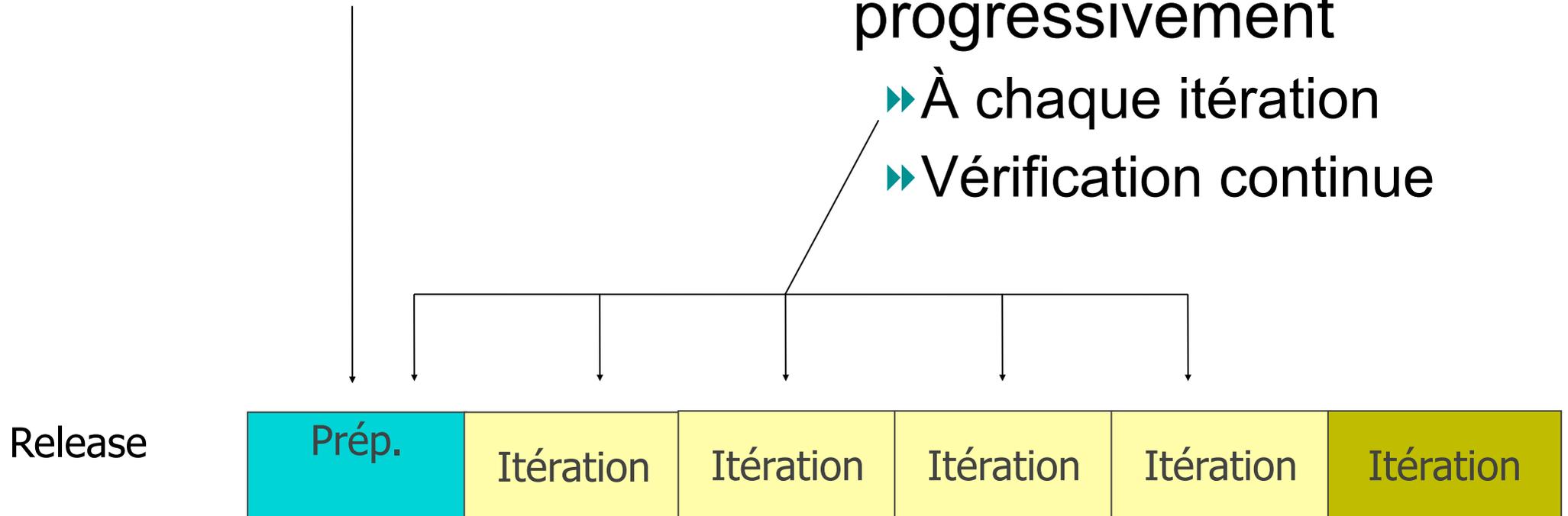


Architecture, 2 approches

- ▶ Concevoir d'abord et coder ensuite

- ▶ Faire émerger la conception progressivement

- ▶ À chaque itération
- ▶ Vérification continue



2 styles d'architecte

▶ Architectus reloadus

- ▶ Prend les grandes décisions
- ▶ Suit les tendances technologiques
- ▶ Ne reste pas toujours dans l'équipe

▶ Architectus oryzus

- ▶ Montre l'exemple en essayant l'architecture
- ▶ Plus orienté code
- ▶ Collabore intensivement avec l'équipe



Intégration continue

1. Compiler, faire le build
2. Tests unitaires
3. Tests d'intégration
4. Rapports d'erreurs

Je crois que toutes les équipes devraient
pratiquer l'intégration continue.
Les outils sont gratuits.
Le seul prix à payer c'est d'apprendre.

Martin Fowler

Une pratique indiscutable



Binôme



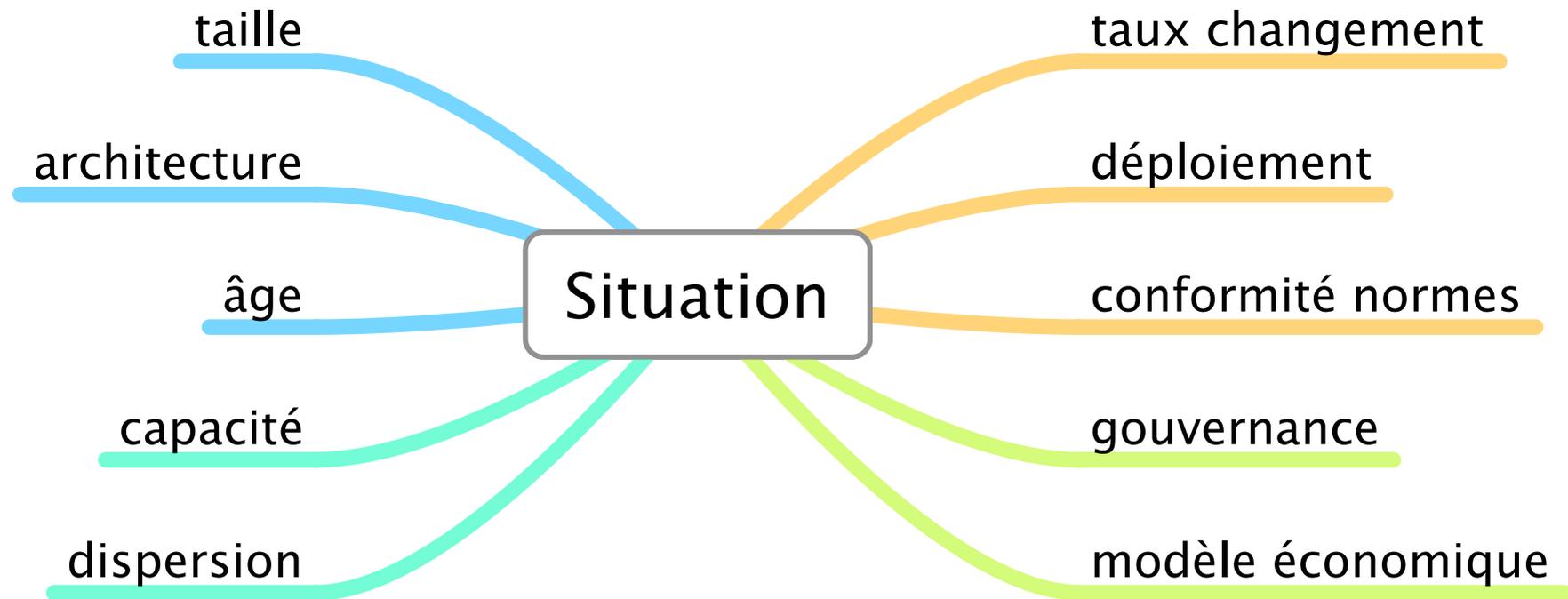
Discuté, mais très efficace
si utilisé avec discernement



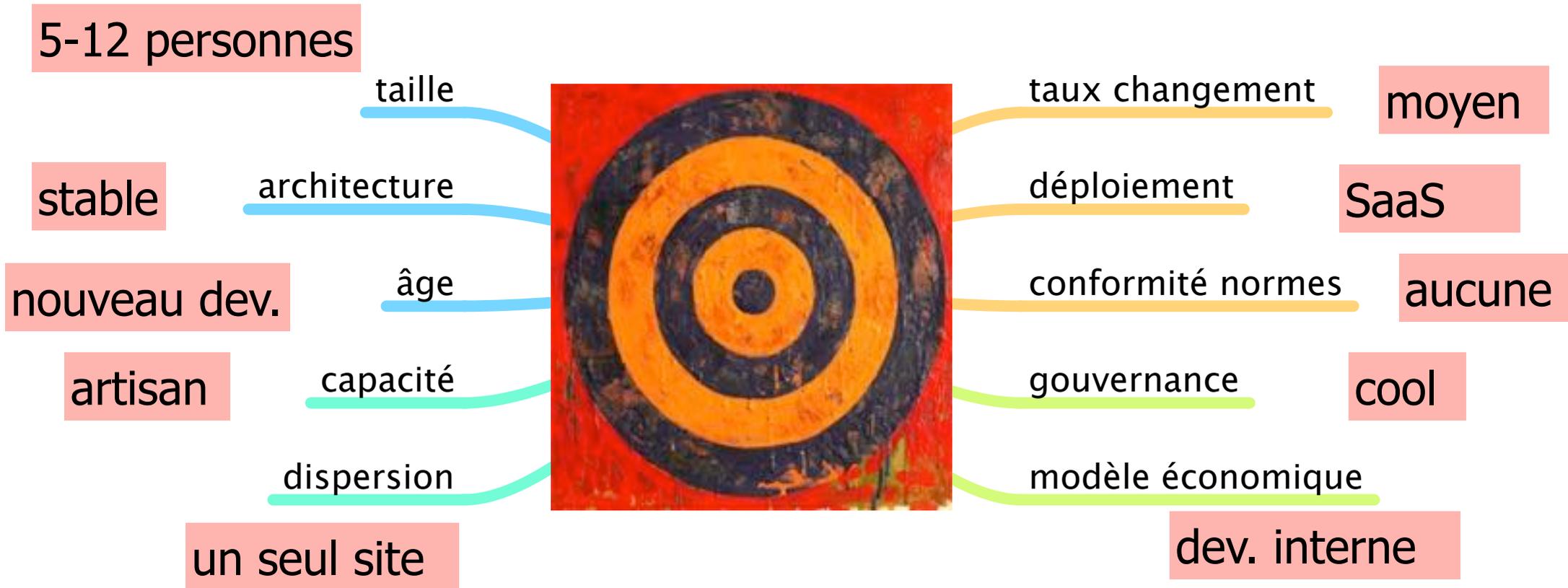
Adapter Scrum au contexte 12



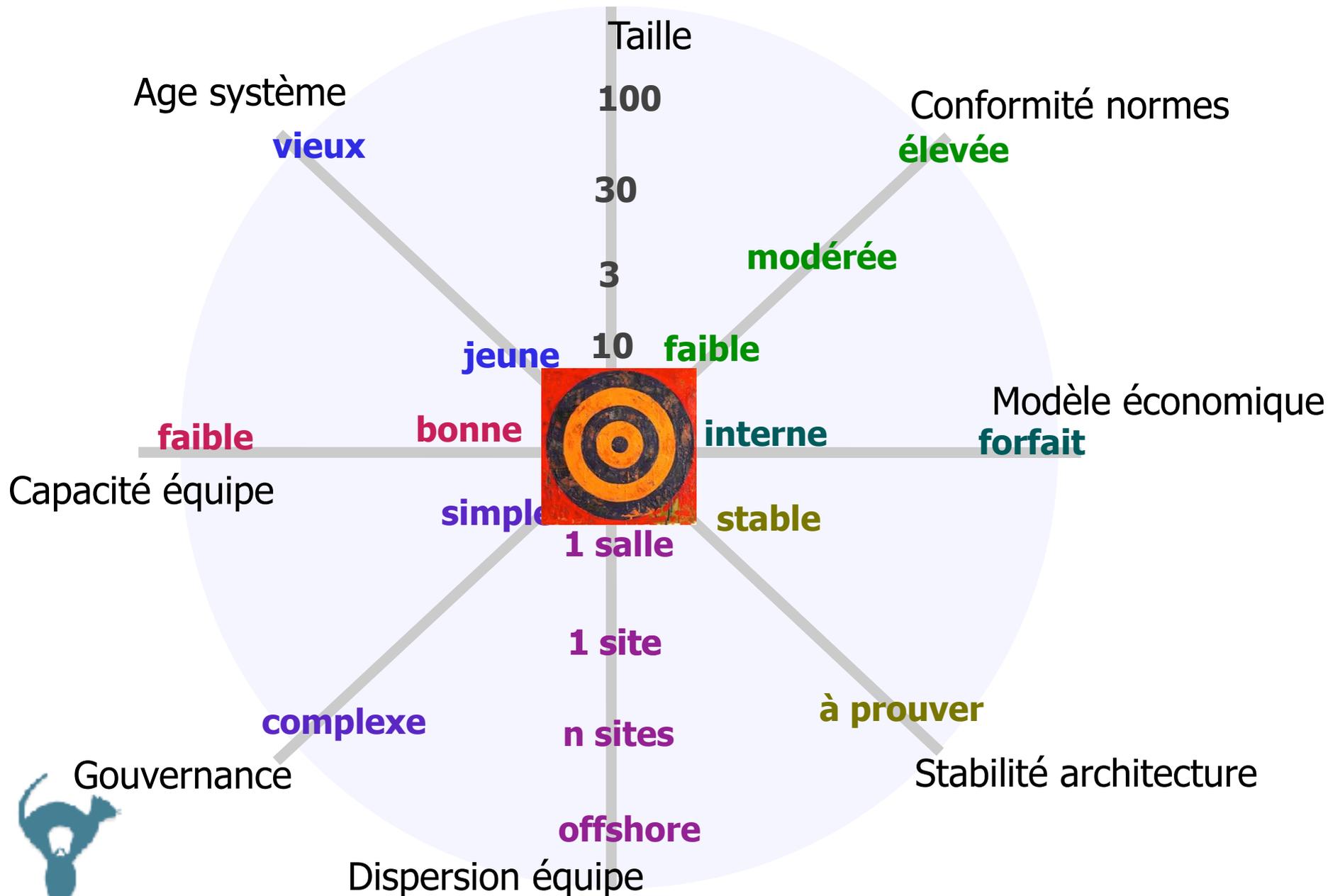
Attributs de contexte



Etes-vous au cœur de cible ?



Evaluation du contexte

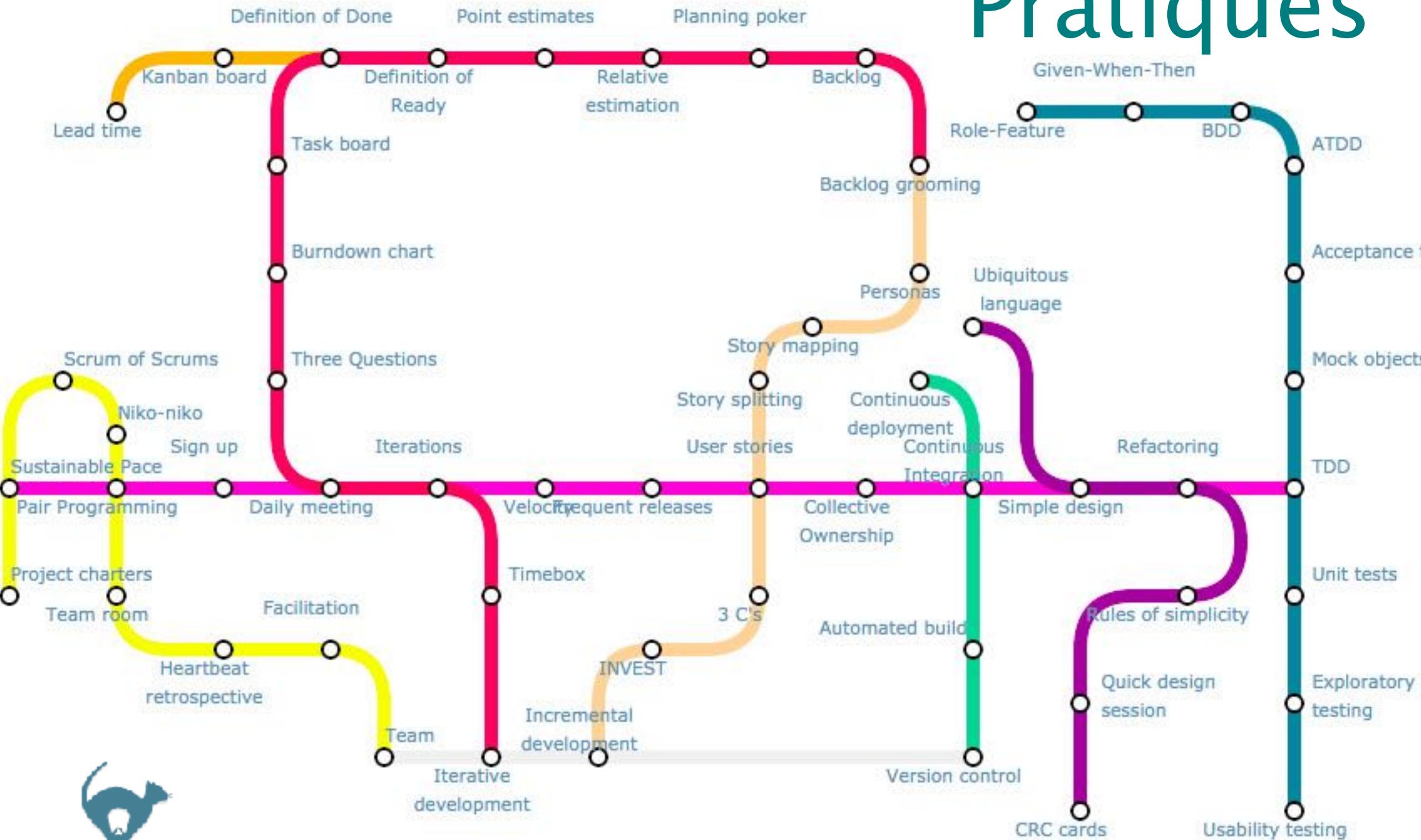


Impact sur les pratiques

		cycle vie	PO	SM team	Backlog	Scrum quot.	Revue	...
taille	S		+	+				
	M							
	L	+	+	+	+	+	+	
	XL	++	+	+	++	+	+	
dispersion	aucune							
	local		+			+		
	nearshore		++	+	+	++	+	
	offshore	+	++	++	+	++	++	
	...							



Pratiques



La transition à Scrum 18



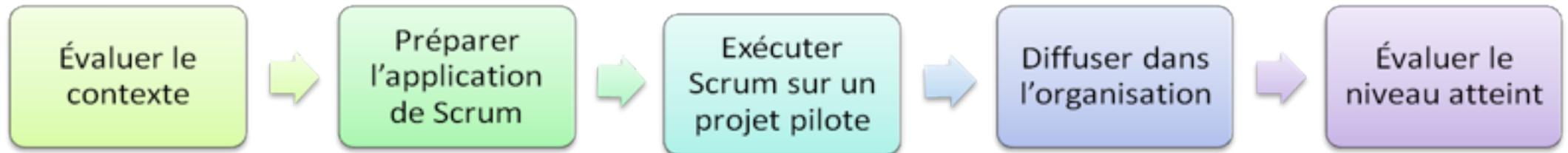
Transition



Comment ?
En combien de temps ?
Avec qui ?



Conseils pour réussir



Modèle de formation



Sensibilisation
1 jour



Application
3 jours



Approfondissement



Extensions du cadre

Scrum

Définition
de produit

Ingénierie
du logiciel

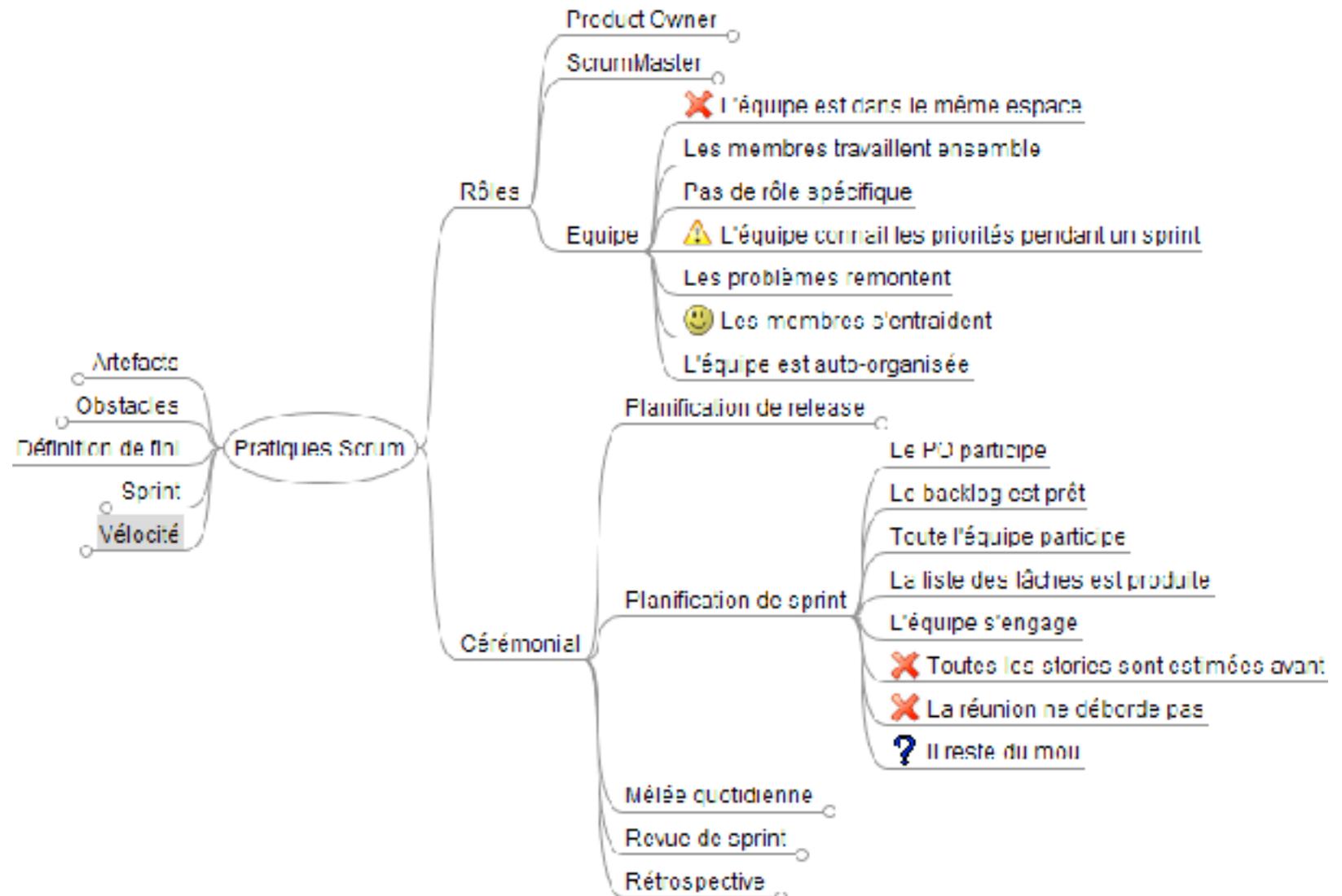
Gestion de
projet



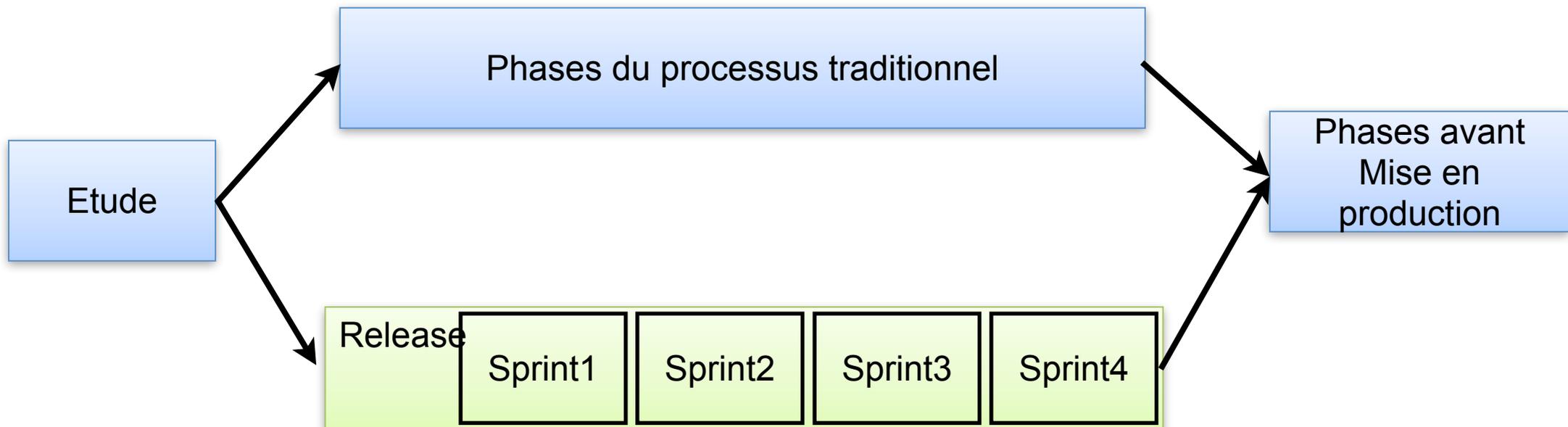
Scrum de scrums



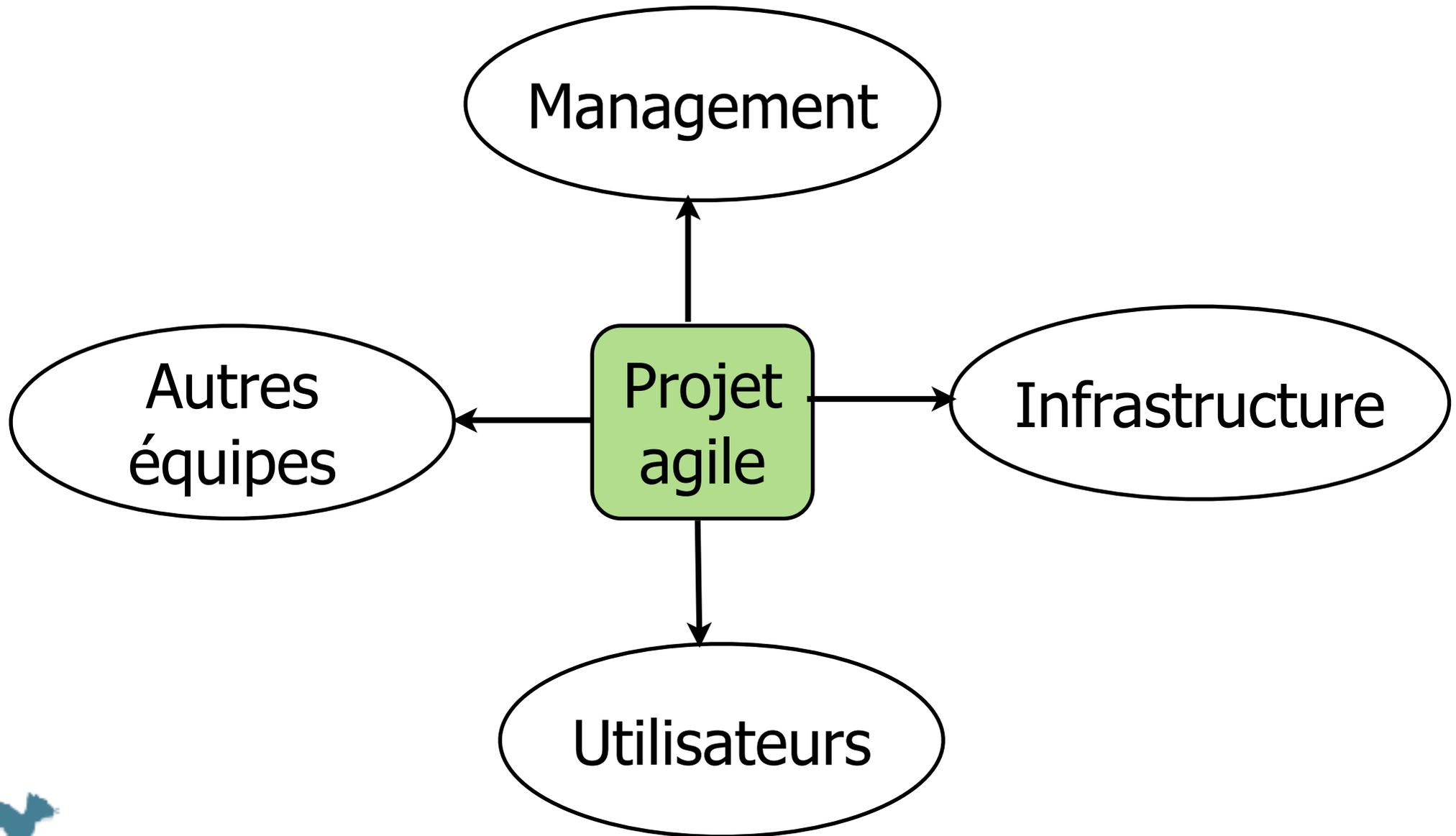
Suivi des pratiques



Mixer ?



Nécessité de diffuser l'Agilité



Scrum d'entreprise



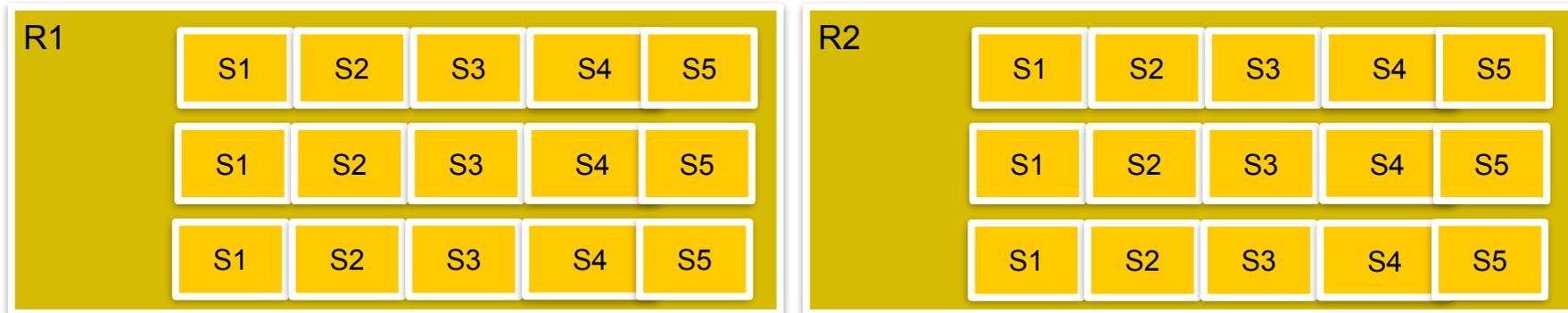
Architectes



Support



Le train de release



Erreurs à éviter lors de la transition

Introduire les méthodes agiles sans un besoin bien compris

Ne pas impliquer tous les intervenants nécessaires

Ne pas saisir toute la portée des changements nécessaires

Ne pas prendre en compte l'aspect social des changements

Sous-estimer la résistance au changement



Conclusion



Ça marche

- ▶ 82% de gain en productivité
- ▶ 77% d'amélioration en qualité
- ▶ 78% de mieux en satisfaction métier
- ▶ 72% de moins en coût de développement

Source: Dr. Dobb's Journal, June 2008

Agilité 1, n'importe quoi
d'autre 0



L'agilité est la capacité d'une organisation à créer de la valeur (*en enchantant ses clients et utilisateurs*), en s'adaptant à temps aux changements dans son environnement

d'après Jim Highsmith
et Philippe Kruchten
(+ Steve Denning)



Demande des efforts

Avoir un appui fort du management

Impliquer toute l'organisation

Améliorer continuellement les pratiques

Ne pas oublier les valeurs et les principes !





L'agilité est un voyage

On s'améliore tout le temps !



Scrum,
Agilité ...et Rock'n roll

