



# Kanban

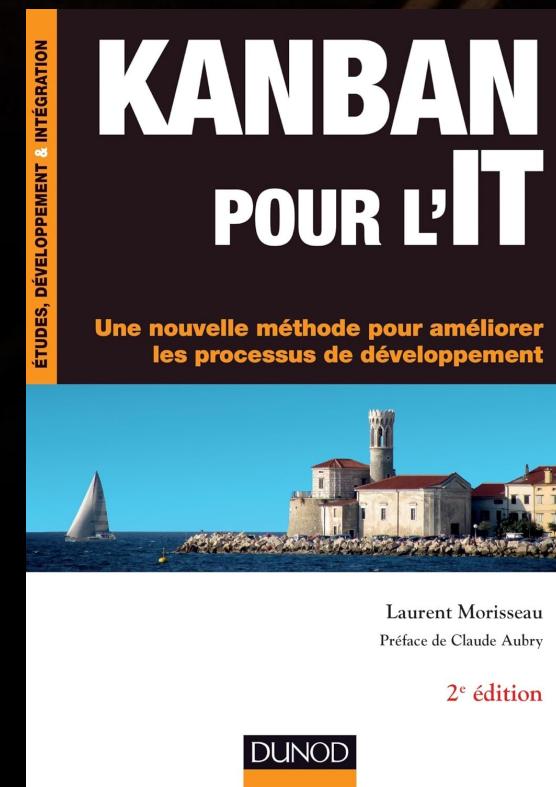
...

## Objectif du cours

### Découvrir le framework Kanban

THE  
KANBAN  
GUIDE

December 2020



<https://kanbanguides.org/english/>

# **Que pouvez-vous me dire sur Kanban ?**

# Orienté flux

Flux

Flux

# Définition

« Kanban est un cadre de gestion agile qui abandonne les itérations au profit d'un travail en flux continu »

## Question :

Qu'est ce que cela signifie concrètement ?

- Avec Scrum nous travaillions avec des itérations de n semaines.
- Ici, il n'y a ni début, ni fin, les sujet arrivent au fur et à mesure. Nous n'avons pas d'itération.

# Objectifs : #1

« Réguler des situations de surcharge de travail ou de flux de demande très variables, afin d'équilibrer la capacité du système à la demande et d'en réduire les variations »

# Objectifs : #2

« Equilibrer le travail planifié du travail non planifié »

**Question :**

Pourquoi ?

- Trouver un compromis entre :
  - Besoins déjà identifiés
  - Demandes internes / externes
  - Les bugs / urgences
  - Les opportunités

# Objectif cible

« Réduire le Time to market »

- L'enjeux de kanban pour une organisation est de réduire le T2M
  - Evoluer avec les concurrents
  - Offrir rapidement de nouvelles fonctionnalités
  - Retour rapide (feedback)

# Comment ?

- En étudiant la fluidité du travail et en :
  - Prenant des décisions au quotidien
  - Répondant localement à des imprévus
  - Optimisant globalement le flux (pas qu'en Dev ...)

# Quand ?

- Kanban répond principalement aux enjeux suivant :
  - travailler sur plusieurs projets en parallèle avec un pool de ressources
  - travailler avec plusieurs équipes spécialisées
    - Pas d'équipe pluridisciplinaire, on se partage des ressources (e.g. UX/UI travaillent avec plusieurs équipes)
  - travailler avec des interruptions fréquentes, support ou urgence à gérer
  - passer au déploiement continu.

# Principles Kanban



# 3 principles

1

Start with what you do now

2

Gain agreement to pursue  
improvement through  
evolutionary change

3

Encourage acts of leadership  
at all levels

# 3 principes : numéro 1

*Commencer avec ce que vous avez maintenant*

- Comprendre le système actuel.
- Respecter les rôles, métiers et responsabilités existantes.
- ⇒ Continuer et améliorer le système actuel

# 3 principes : numéro 2

*Acceptez d'appliquer des changements progressifs*

- Introduire des améliorations progressives plutôt que de mettre en œuvre des changements brutaux en une seule fois.
- Les petites améliorations progressives sont souvent plus faciles à gérer, plus durables et plus faciles à adapter pour les équipes.
- Kanban encourage les équipes à évaluer régulièrement leurs processus, à identifier les domaines à améliorer et à procéder à des ajustements progressifs.

# 3 principes : numéro 3

*Encourager les actes de Leadership à tous les niveaux*

- Une équipe auto-organisée et auto-responsable

# 6 pratiques maitresses

1. Visualiser le flux de travail
2. Limiter les travaux en cours (WIP)
3. Gérer et mesurer le flux de travail
4. Rendre explicite les règles de gestion de processus
5. Implémenter des boucles de feedback
6. Améliorer en collaboration

# « Utilisation » de Kanban

« Le cadre Kanban n'impose pas l'utilisation d'itérations divisées en blocs de temps »



Le cadre Kanban nous dit uniquement de gérer le *flow of work* et le *work in progress*



Pas de périmètre fixe

# Vocabulaire Kanban

Terminologie

Terminologie

Terminologie

Terminologie

Terminologie

Terminologie

Terminologie

Terminologie

Terminologie

## Question :

Que voyez-vous ?

tableau

WIP

Pending

3

Development

5

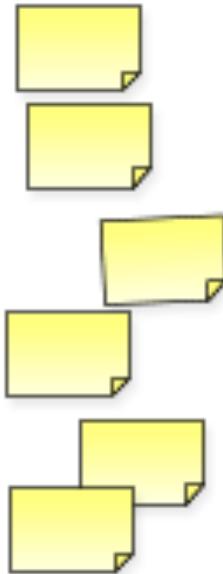
Test

3

Deploy

5

Cartes



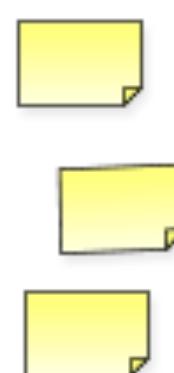
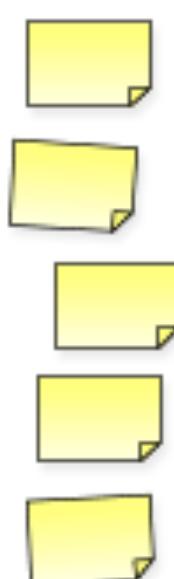
Doing

Done



Doing

Done



Débit

Le temps de traversé

# Quel type de management est utilisé ?

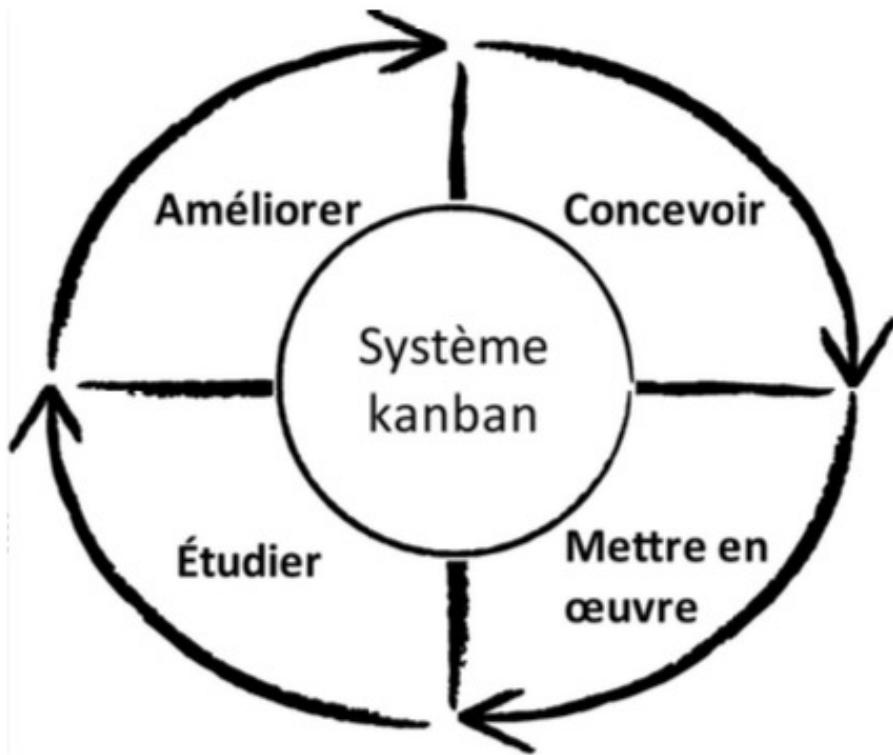
Management visuel

Tableau, cartes

# La démarche Kanban



# Basé sur Plan-Do-Study-Act (PDSA)



## Objectif :

Mieux comprendre les relations de cause à effet  
ET identifier les ajustements à faire

# 1. Concevoir

Question :

Pour concevoir il faut d'abord ?

Le concepteur du système Kanban doit :

- Identifier le processus existant
- Les éléments de travail
- Les règles

Question :

Avec qui ?

Réalisé avec l'équipe et les représentants des parties prenantes

## 2. Mise en oeuvre

Question :

Que cherche-t-on à faire

L'équipe se cherche et essaie de maîtriser le système mis en place

- Daily
- Mise à jour du tableau
- Indicateurs de suivi
- ...

=> EMPIRISME

### 3. Etudier

L'équipe Kanban étudie le comportement et les réponses du système au règles établies lors de la phase de conception. Pour ce faire, on peut s'appuyer sur différent modèles :

- théorie des files d'attente pour optimiser le flux de travail
- théorie des contraintes pour identifier et traiter les goulot d'étranglement
- le gaspillage Lean pour réduire les délais de blocage

## 4. Améliorer

- L'équipe tire les enseignements du cycle qui se termine.
- Elle ajuste le système, les règles en conséquence.
- Pour s'assurer que l'amélioration apportée au système soit mise en place, l'équipe :
  - partage les règles de manière explicite
  - simplifie là où c'est possible
  - met à jour les standards
  - valide les micro-ajustements fait au quotidien

#1  
Concevoir

A dark, atmospheric night scene. In the upper right quadrant, a tall street lamp stands with two rectangular lights glowing brightly. To its left, another smaller street lamp is partially visible, also with its lights on. The rest of the scene is engulfed in deep shadow, creating a moody and minimalist atmosphere.

Concevoir

# Un travail collaboratif

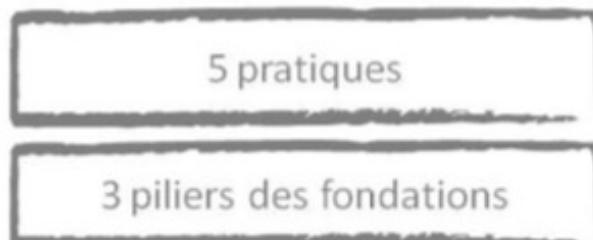
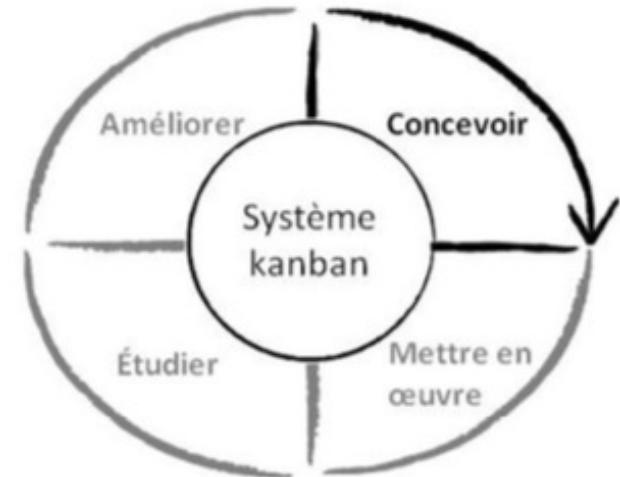
## Affirmation

La conception du système Kanban est un travail collaboratif entre le concepteur, l'équipe et les parties prenantes

## Question :

Pourquoi ??

- Pour que chacun puisse se l'approprier et le faire évoluer (phase étudier et améliorer)



## Système kanban

Portée et objectifs

Nature de la demande

Capacité du système

Éléments de travail

Flux de travail

Flux tiré

Juste à Temps (JAT)

Carte kanban

Tableau kanban

Limite haute

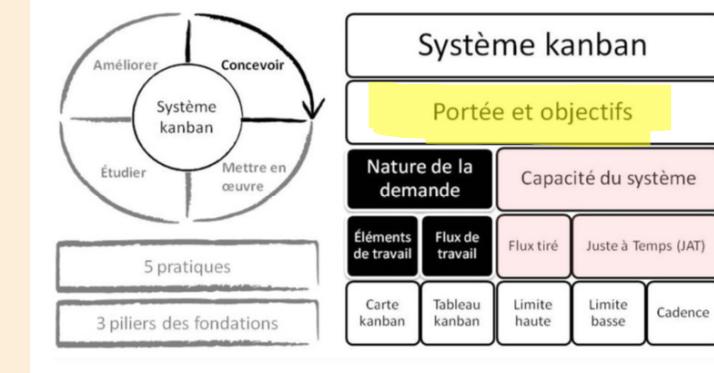
Limite basse

Cadence

# La portée

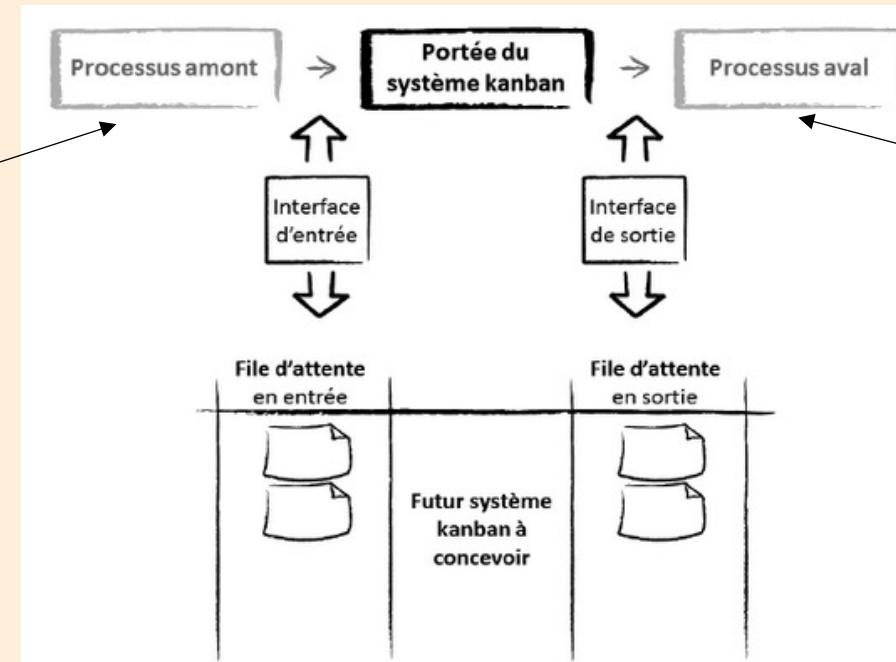
## Affirmation

**Le système kanban commence et s'arrête là où les équipes peuvent être propriétaire du processus**



Ne maîtrise pas

Ne maîtrise pas



**Figure 6.6** — Les interfaces du système kanban.

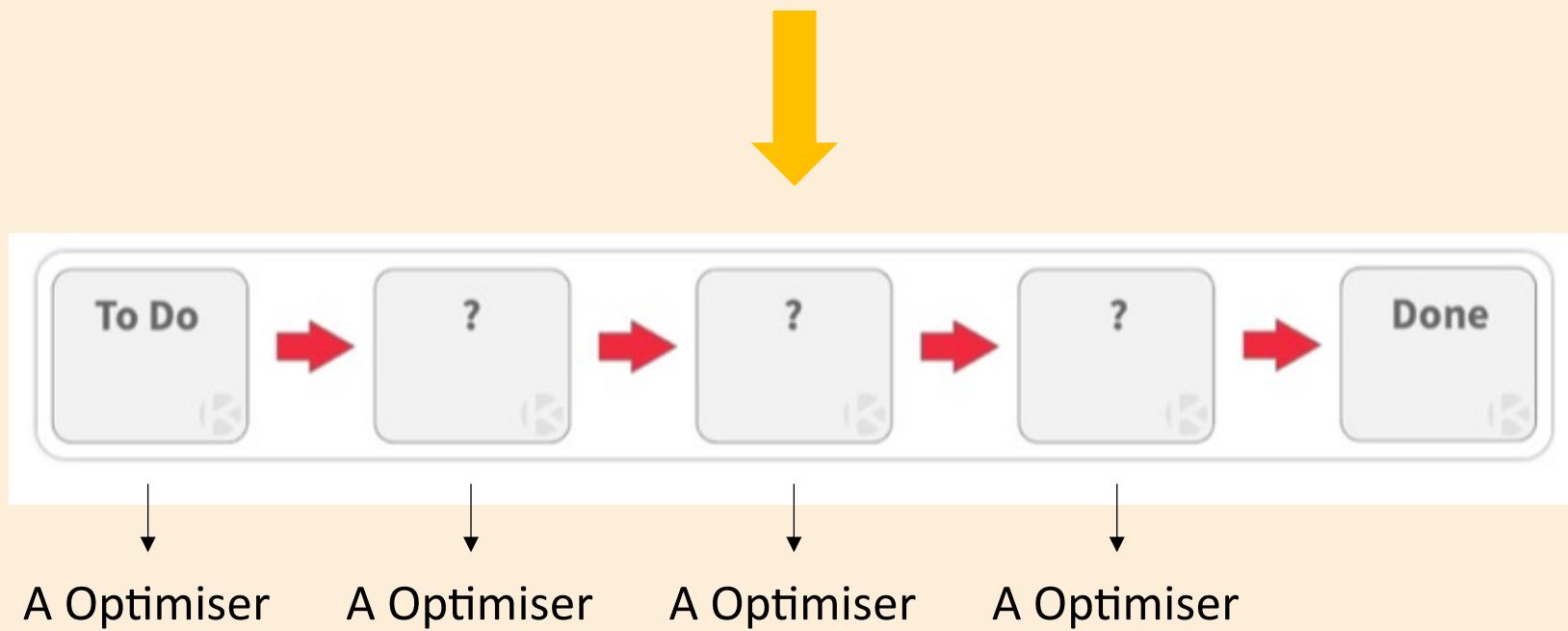
# Nature de la demande

Nature de la demande		Capacité du système		
Éléments de travail	Flux de travail	Flux tiré	Juste à Temps (JAT)	
Carte kanban	Tableau kanban	Limite haute	Limite basse	Cadence

- Définir
  - L'élément de travail
  - Le flux de travail

# Objectif

«Le problème à résoudre est le workflow, la solution consiste à le visualiser (carte + Kanban Board) »

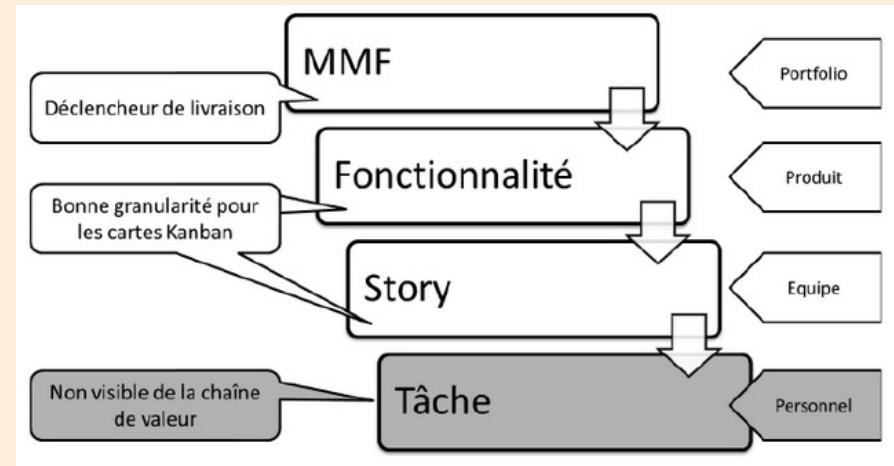


# Elément de travail

Un élément de travail peut être :

- Une US
- Une ticket d'incident
- Un ordre de travail pour les opération
- ...

Catégoriser les types d'éléments  
=> e.g WIP par catégorie



« On peut visualiser un élément de travail via une carte Kanban »

Nature de la demande	
Éléments de travail	Flux de travail
Carte kanban	Tableau kanban

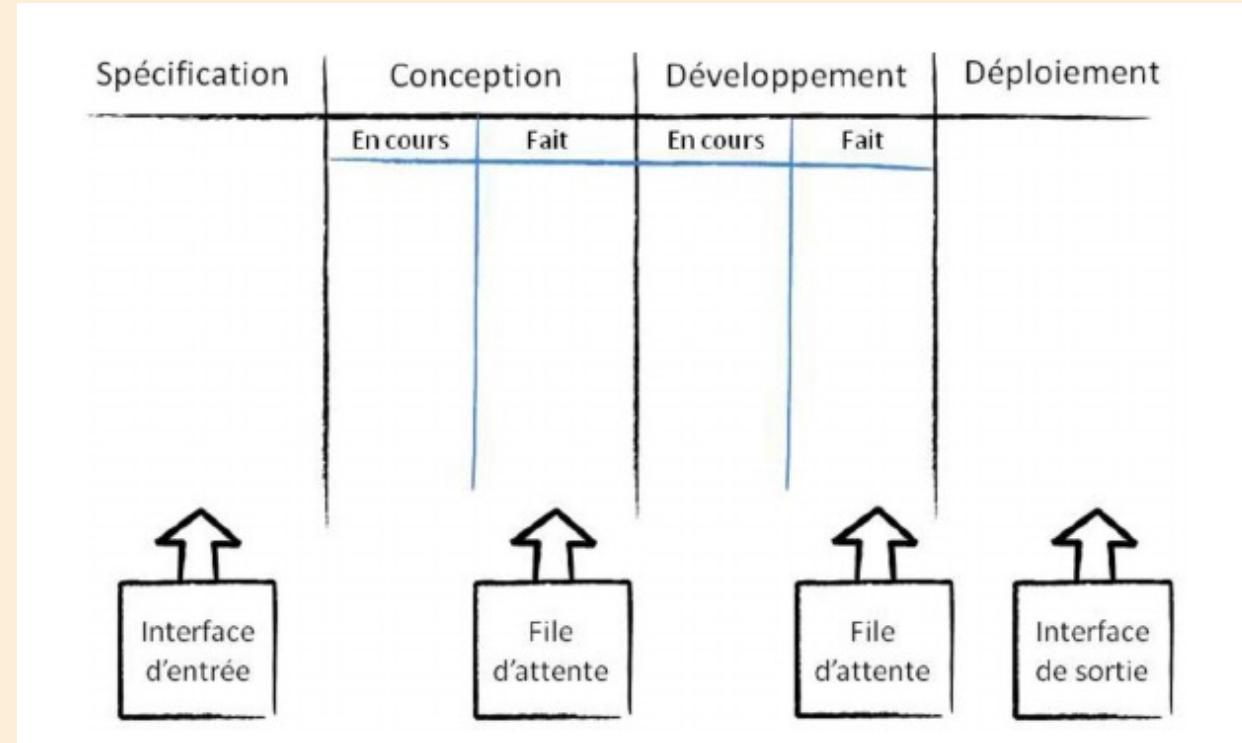
# Elément de travail

Une carte contient assez d'information pour permettre le travail en flux :

- Date d'entrée
- Date de sortie réelle (une fois effectuée => calcul vitesse)
- Identifiant
- Description (e.g. Give-Then-When)
- Critère d'acceptation



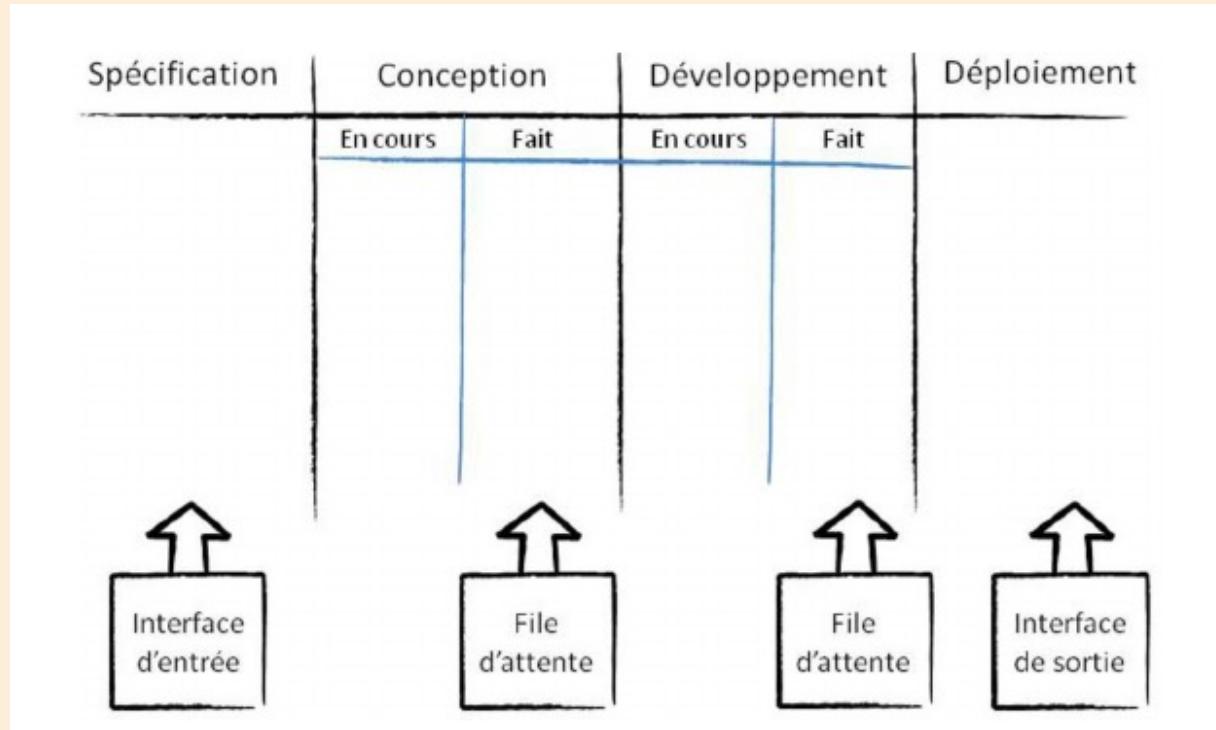
# Visualiser le flux de travail



**Objectif :** suivre les *éléments de travail* tout au long de la chaîne

- Visualiser le travail accompli
- Identifier les dysfonctionnement (e.g. goulot d'étranglement)

# Visualiser le flux de travail



**Interfaces** : c'est un buffer qui représente des éléments (carte) prêts (pour la conception ici) mais qui ne peuvent être encore *en cours* car il n'y a pas de ressource disponible (wip)

**File d'attente** : sépare les équipes spécialisées pour leur permettre de travailler de manière asynchrone.

# Des kanban dans Scrum ?!

- On voit souvent des kanbans dans Scrum
- Parce que la théorie Scrum valorise la transparence et la visibilité pour faciliter le cycle d'inspection et d'adaptation.
- Les tableaux aident à visualiser le travail, et cette pratique est donc couramment empruntée à Kanban.

⇒ Le tableau est juste une aide visuelle.

# Un management visuel

- Carte + tableau => management visuel

Question :

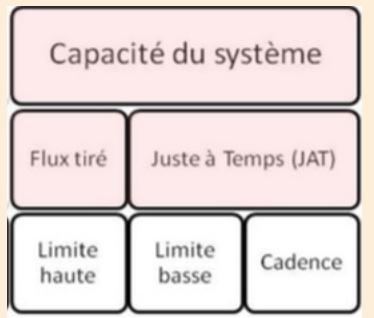
Pourquoi ce type d'approche est intéressant

- il influence le comportement et l'attitude des individus et du groupe
- il aide à prendre les bonnes décisions
- il rend visible le gaspillage
- il constitue, en soi, un des moteurs de l'amélioration continue.

# Capacité du système

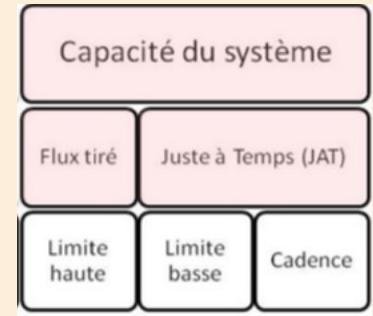
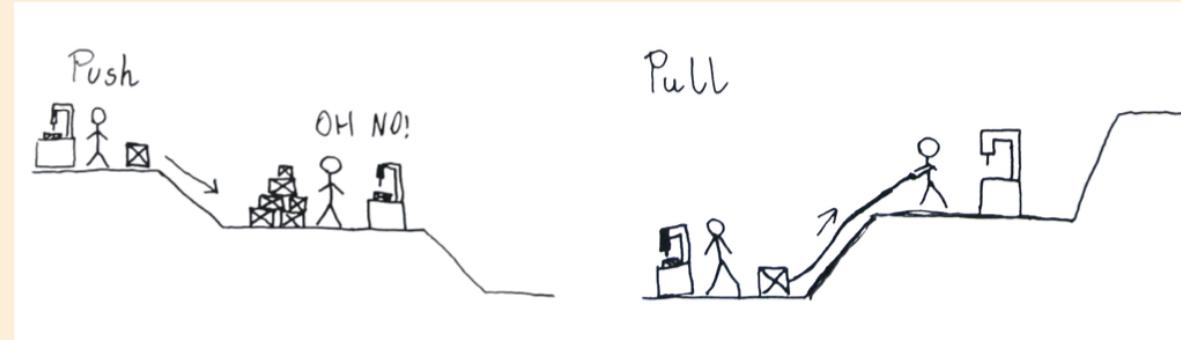
Nature de la demande		Capacité du système		
Éléments de travail	Flux de travail	Flux tiré	Juste à Temps (JAT)	
Carte kanban	Tableau kanban	Limite haute	Limite basse	Cadence

# Flux tiré

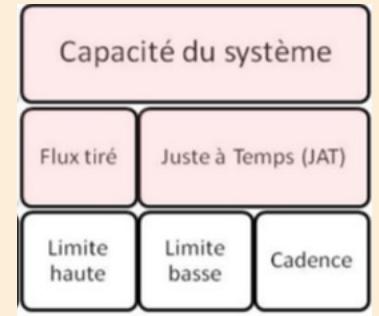


« Dans un système "pull" » (tiré) une tâche est récupérer uniquement lorsque l'équipe à la capacité de la gérer. »

# Flux tiré



Flux poussé	Flux tiré
Les tâches sont créées, planifiées avant qu'on en ait besoin	Les tâches sont créées et priorisées en fonction du besoin existant
Les tâches sont assignées aux personnes	L'équipe choisit la tâche prioritaire dès qu'elle a la capacité de la traiter
Le manager peut estimer la vitesse d'une personne	Pour commencer une nouvelle tâche il faut finir la précédente (WIP max)
	Réduit le surplus de tâches
Scrum push et pull Push : travail défini en avance (Sprint Planning) Pull : limite de travail max == le sprint	Kanban que pull L'équipe dépile la charge de travail



# Flux tiré : limite hausse (WIP)

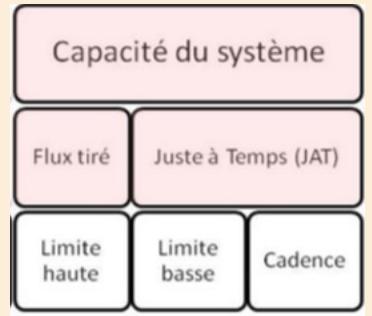
« Le *Work in progress* (limite de travail) définit la quantité maximale de travail qui peut exister dans chaque colonne. »

## Question :

Que permet le WIP ?

- Trouver les goulots d'étranglement
- De visualiser les points bloquants
- Prévoir mouvements des ressources humaines si on va rencontrer un goulot

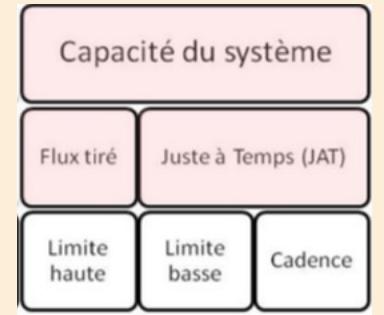
# Flux tiré : limite hausse (WIP)



Question :

Le WIP permet d'éviter ?

- Sur-engagement : on a une capacité max on peut pas s'engager au delà
- Multitâches
- Sur-production d'une équipe par rapport à la capacité du système
  - (E.G. limite max d'une file d'attente => aller aider d'équipe de droite)

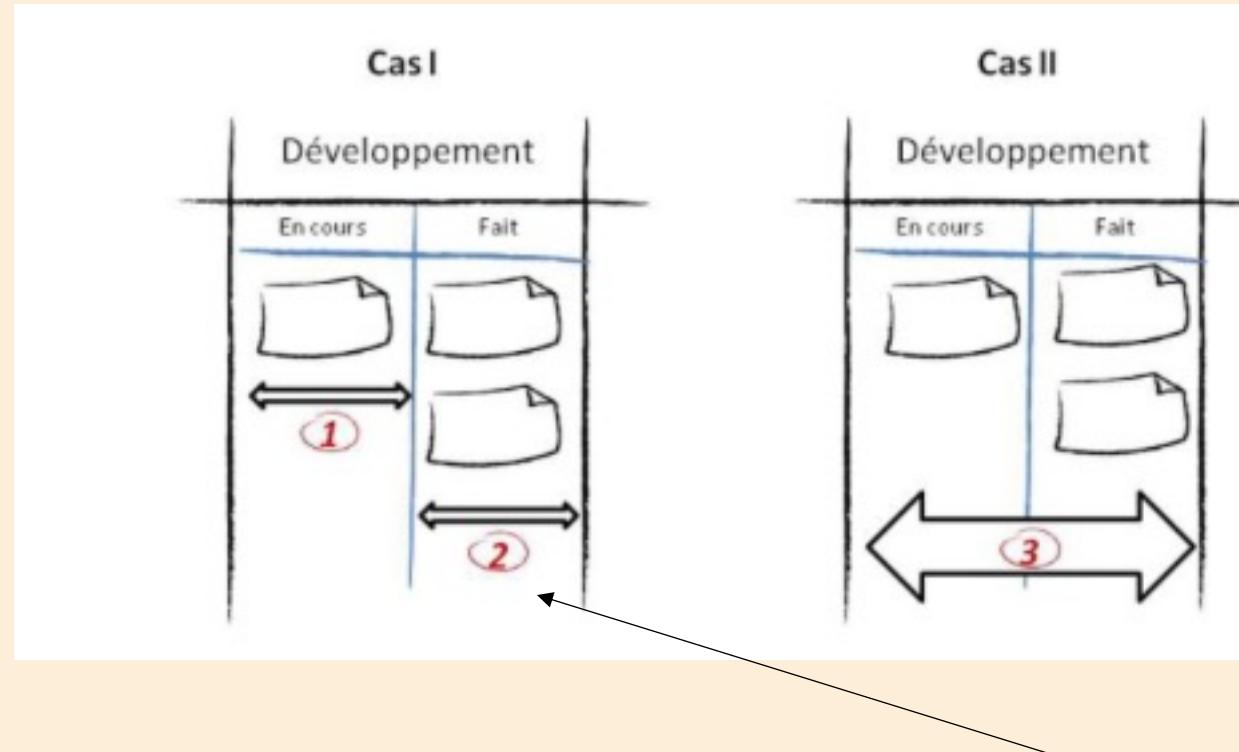


# Flux tiré : WIP par type (optionnel)

Si WIP = 10

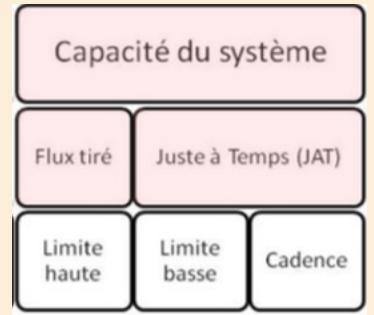
- Anomalie : 10% => max 1 évo dans la colonne
- Evolution : 70% => max 7 évol dans la colonne
- Review : 20% => max 2 évol dans la colonne

# Flux tiré : WIP précis

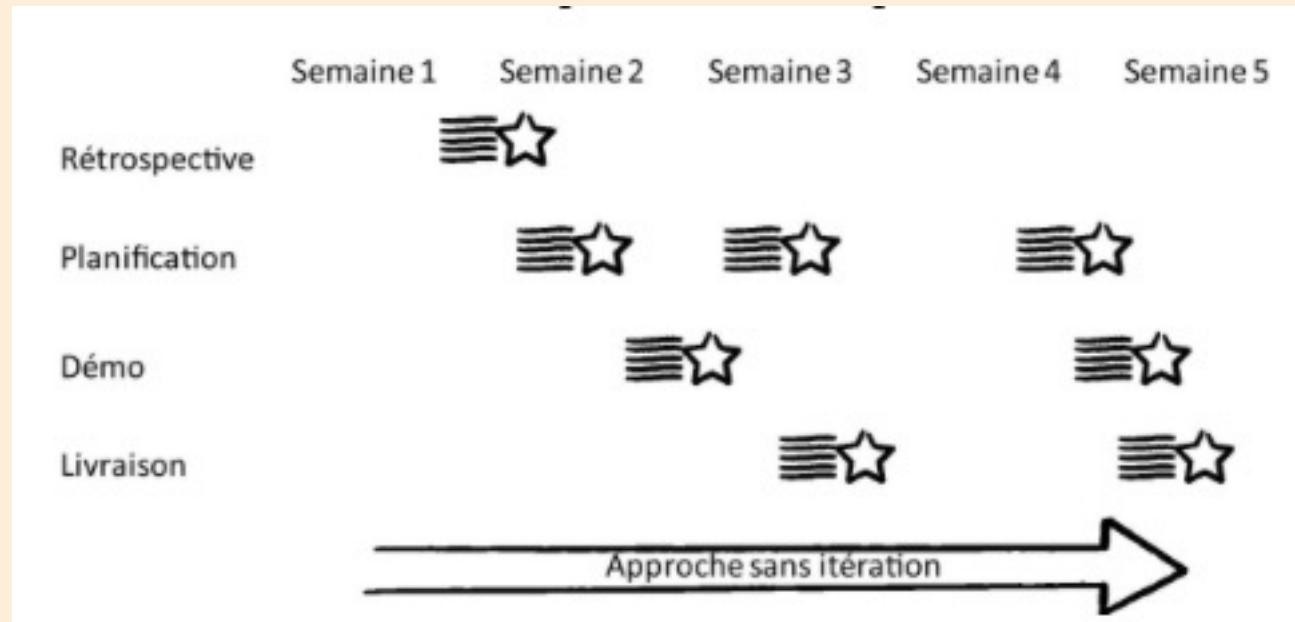


Proposition : Mettre une WIP sur une file d'attente

# Juste à temps



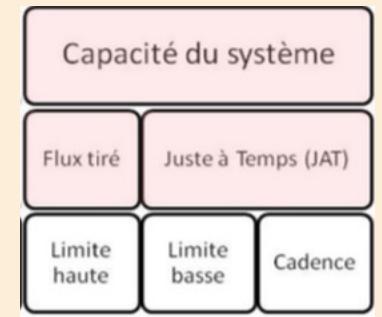
« Avoir une planification JAT lissée sur la demande. L'objectif est de réduire le temps de traversée en réduisant les en-cours et les files d'attente »



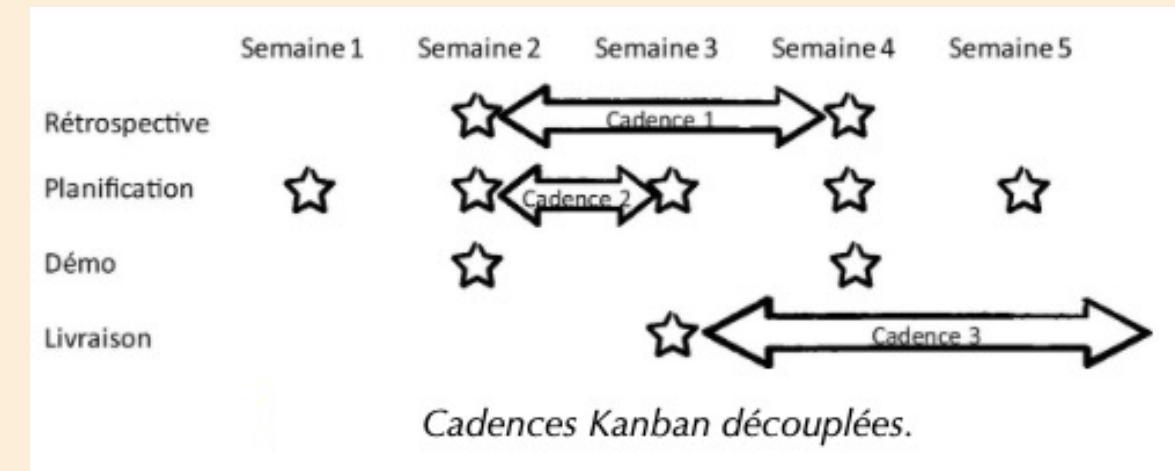
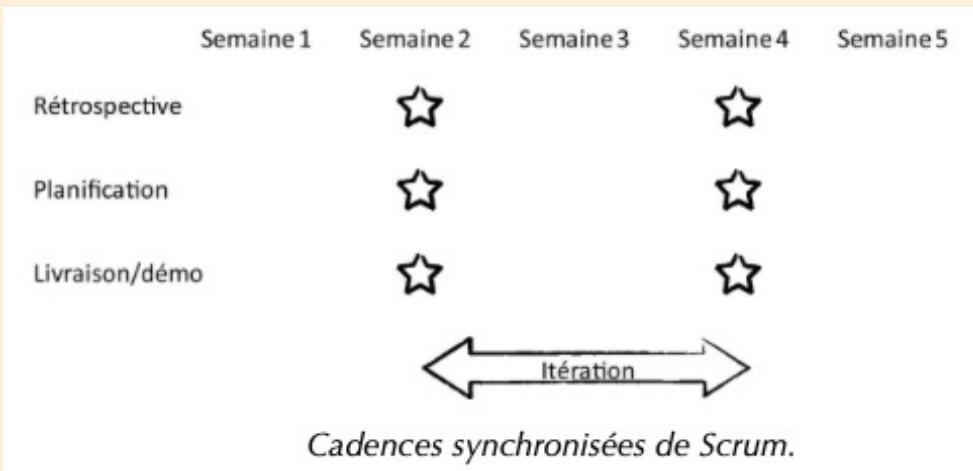
- Repousser la prise de décision
- Traiter à la demande
  - Injection d'élément de travail à la demande
  - Livraison dès que release prêté

JAT sert à réduire le temps entre une prise de décision et l'effet qu'elle va produire

# Juste à temps : cadence

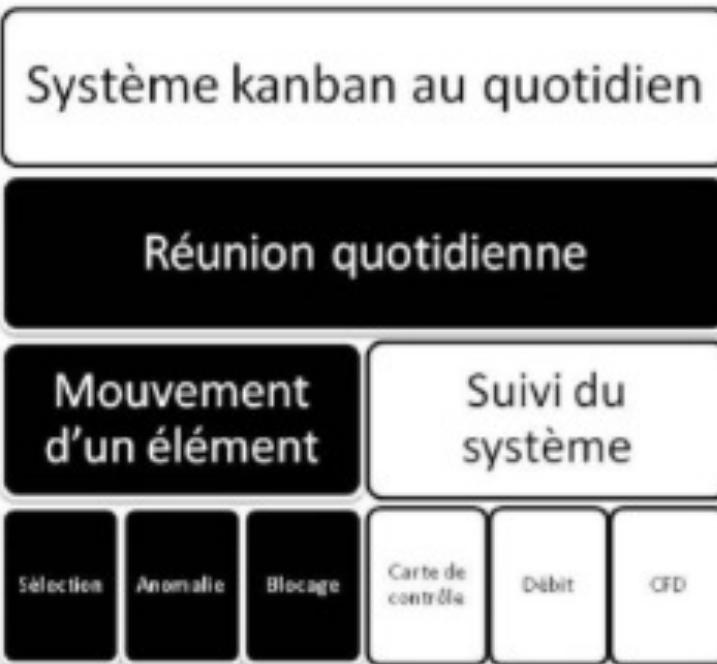
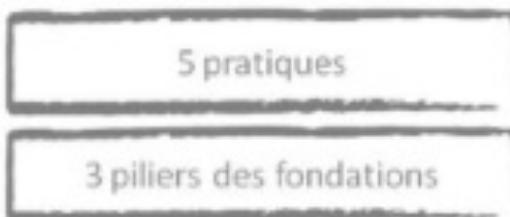
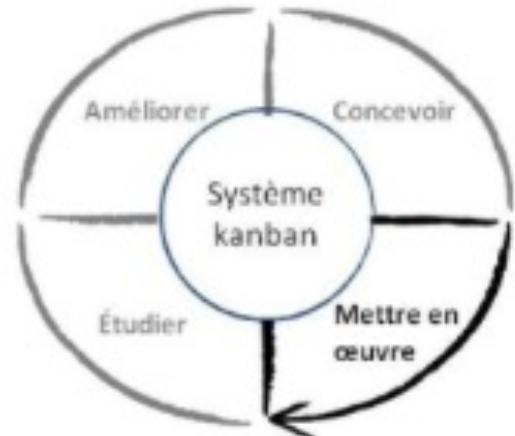


« Chaque activité peut avoir un cadence différente des autres »



#2

Mise en place



# Mouvement d'un élément



- Gérer le mouvement
- Gérer l'affection
- Gérer le blocage
- Gérer anomalie

# Gérer le mouvement

- Chaque jour l'équipe se regroupe autour du *Kanban Board* où elle va organiser son travail pour la journée à venir

**Question :**

Que fait-on durant cette réunion ?

- Mise à jour du tableau ?
- Qu'est qui est le plus important à faire aujourd'hui ?
- Qui travaille sur quoi ?
- A-t-on des blocage ?

# Gérer le mouvement 2

- Les membres de l'équipe choisissent l'élément sur lequel ils veulent travailler (oui et non; slide suivante)
- **Pas de priorisation en amont; prise de décision sur le moment**
  - « Avec Scrum la priorité est donnée par le PO » (atteindre à minima le SGoal)
  - Ici être stratégique : choisir le « bon » type pour ne pas bloquer le système, le plus rapide à faire et le plus prioritaire

# Gérer l'affection

Question :

Quand est faites l'affection ?

- Affection faite au dernier moment

Question :

Pourquoi ?

Affecter trop tôt contribue à l'accumulation du travail  
(stock => lean)

Question :

Comment ?

L'affection se fait au mieux POUR optimiser le flux de travail

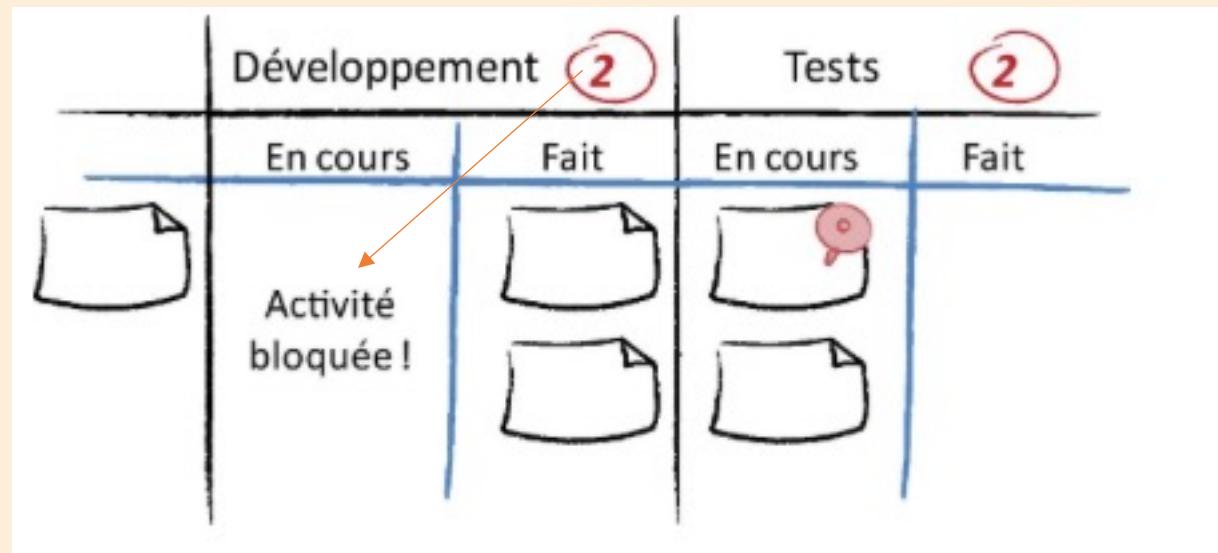
- On vise l'optimisation globale
- Travailler en fonction de ses compétences afin de fluidifier le système

# Gérer le blocage

« Il y aura régulièrement des blocages »

WIP max == 2

Or 2 tâche en FAIT



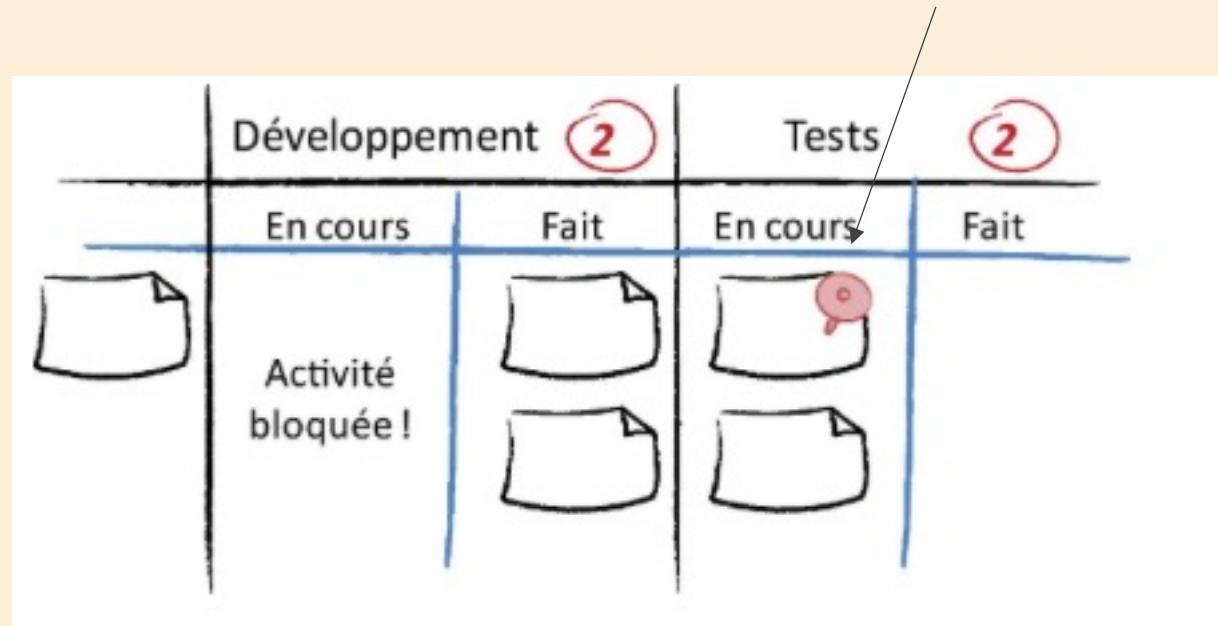
Les devs attendent que le système se débloque ??

# Gérer le blocage : répondre par la demande

- Ne pas commencer un travail si on sait qu'il va être bloqué
- Débloquer le système en travaillant juste assez. On se demande quel est le minimum à réaliser pour faire avancer l'item
- Ne pas accepter d'autres demandes

# Gérer le blocage : répondre par la capacité

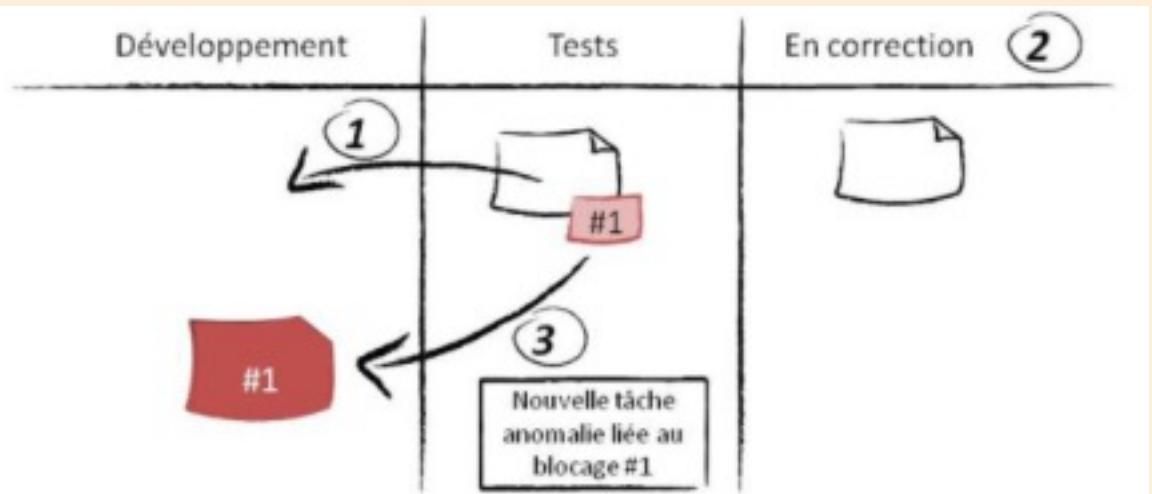
- Je suis bloqué
- Je vais proposer mon aide en aval pour débloquer le système



WIP Dev/Fait maximum atteint  
=> Je vais dans l'équipe de tests

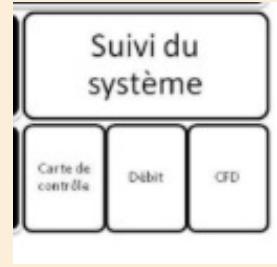
# Gérer les anomalies

- Que faire si une carte kanban se retrouve bloquée (pas d'expert, trop complexe) ?
1. Est-ce quelle revient en arrière ?
  2. On crée une colonne en correction ?
  3. Créer un nouvelle tâche : anomalie (permet de tracer)



# Suivi du système



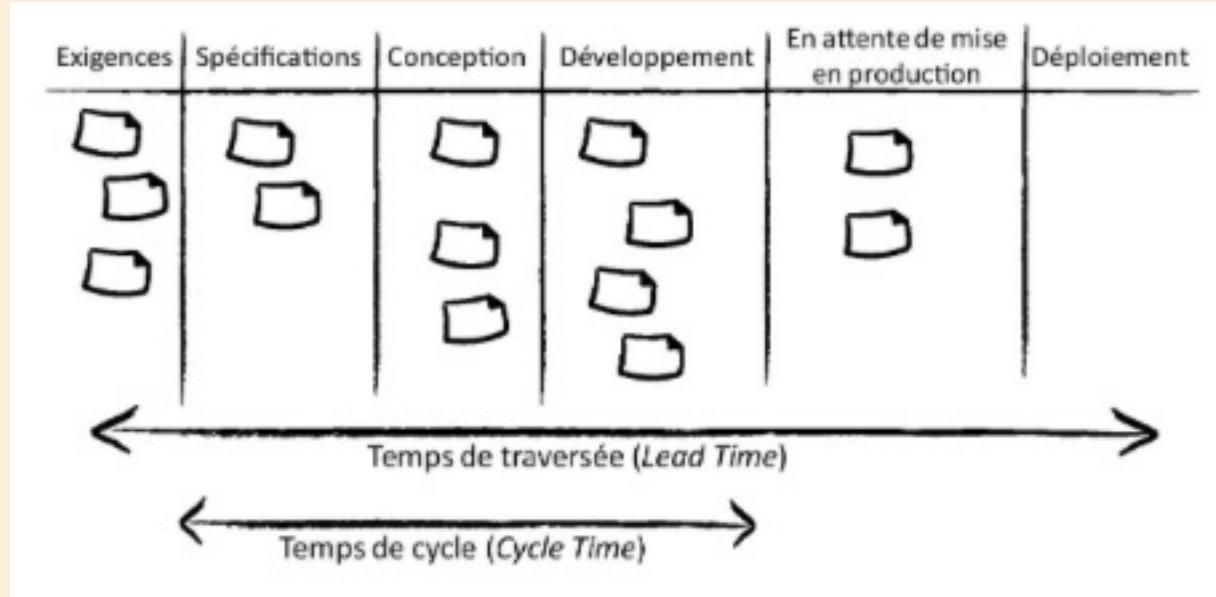


# Mettre à jour les indicateurs

« Lors des réunions quotidienne l'équipe met à jour les informations du tableau et des cartes kanban »

- Le temps de traversé (temps de cycle / cycle time)
- Le débit
- Le nombre d'élément par activité

# Suivre le temps via les cartes



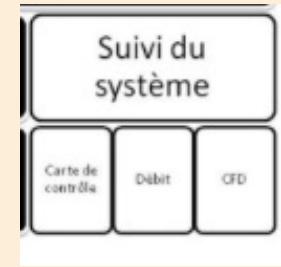
## Lead time :

- Backlog jusqu'à la livraison
- Obj : le réduire car c'est ce que voit le client

## Cycle time :

- IN PROGRESS jusqu'à « produit fini »
- Optimiser notre cycle time pour réduire le Lead Time

L'équipe est maître du cycle time (son kanban), elle doit donc l'optimiser



# Suivre le débit

**Débit correspond :**

« *On nombre d'éléments sortant du système par unité de temps* »

Scrum : Vélocité == débit de stories points par sprint

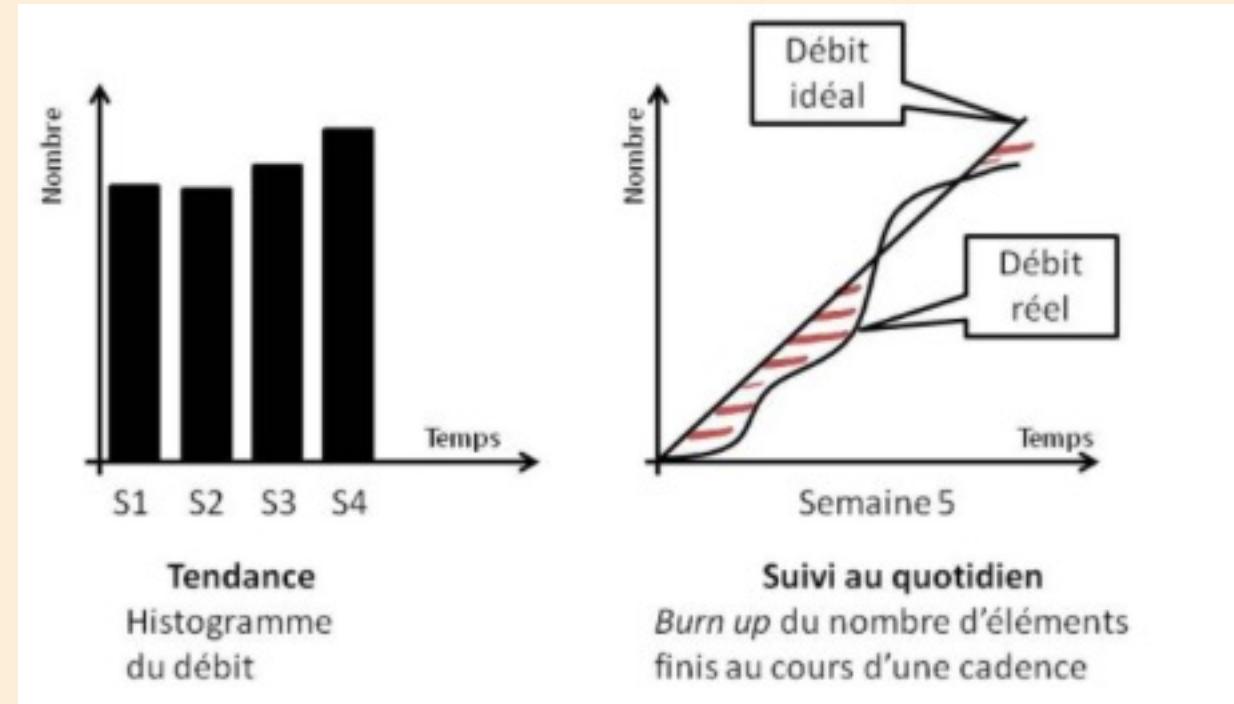
Kanban : Débit == début d'US sortante par temps

(Pas de sprint)

**Question :**

Quel outil ? (slide suivante)

# Suivre le débit : burnup chart

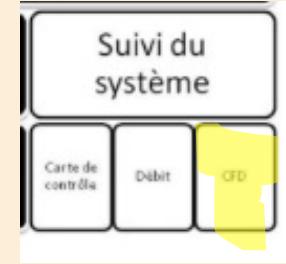


On représente le nombre d'éléments finis (SP) en fonction du temps

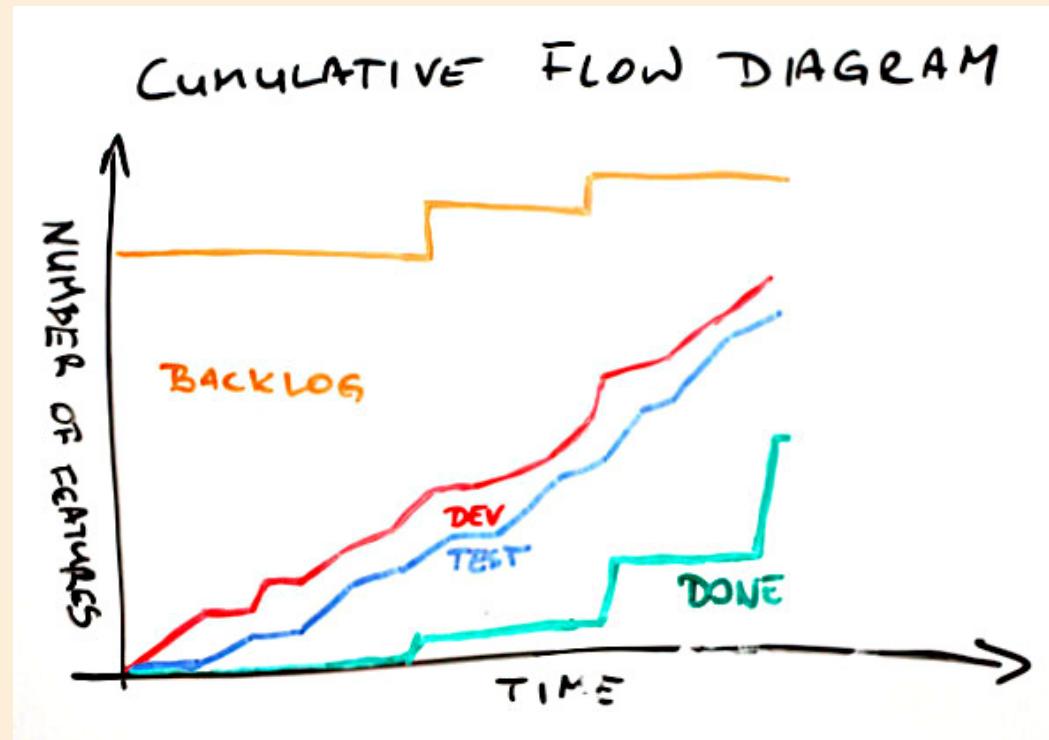
TD burnup chart

Puis on compare la courbe réelle à la courbe idéale

# Suivre le nombre d'éléments dans le système : CFD



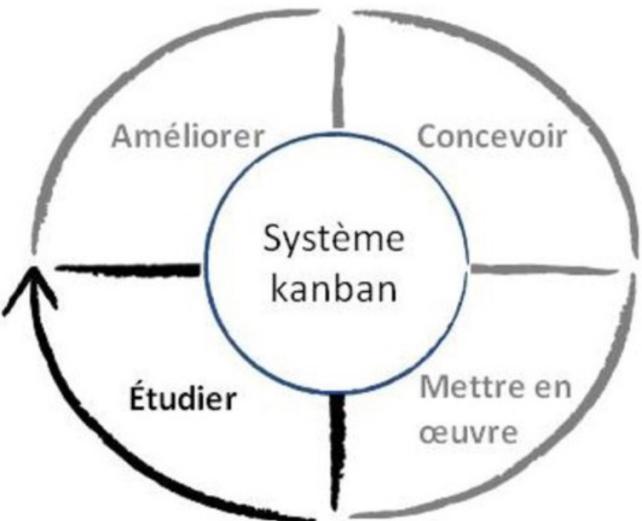
Suivre le nombre d'élément en cours par activité



TD cdf

#3

Etudier



5 pratiques

3 piliers des fondations

## Étude des dysfonctionnements

### Système globalement saturé

- Théorie des files d'attente
- Ajuster la capacité

### Système localement saturé

- Théorie des contraintes
- Goulet d'étranglement

### Système non saturé mais variable

- Maîtrise statistiques des procédés
- Limites de variabilité

### Limites trop hautes vs capacité

- Gaspillage du Lean
- Diminuer les limites

# Etudier un système Kanban

*Se poser des questions :*

- Les règles du système kanban ont-elles pu être suivies ?
  - Les difficultés ou dysfonctionnements générés par la mise en place de limites sont riches d'informations pour comprendre le système.
- Si c'est le cas, le système est-il sous contrôle ?
  - Il s'agit de valider les hypothèses de départ. L'équipe s'assure qu'il va être possible de définir la référence de capacité du système.
- S'il est sous contrôle, est-il optimisé ?
  - Y a-t-il des opportunités d'amélioration du système et de sa capacité ? »

# Etudier un système Kanban : outils

- Théorie des contraintes
  - Donne des pistes pour travailler avec des goulot d'étranglement
- Chasse au gaspillage (LEAN)
  - Value Stream Mapping qui aide à réduire les limites sur les activités

# Globalement saturé

## Étude des dysfonctionnements

### Système globalement saturé

- Théorie des files d'attente
- Ajuster la capacité

### Système localement saturé

- Théorie des contraintes
- Goulet d'étranglement

### Système non saturé mais variable

- Maîtrise statistiques des procédés
- Limites de variabilité

### Limites trop hautes vs capacité

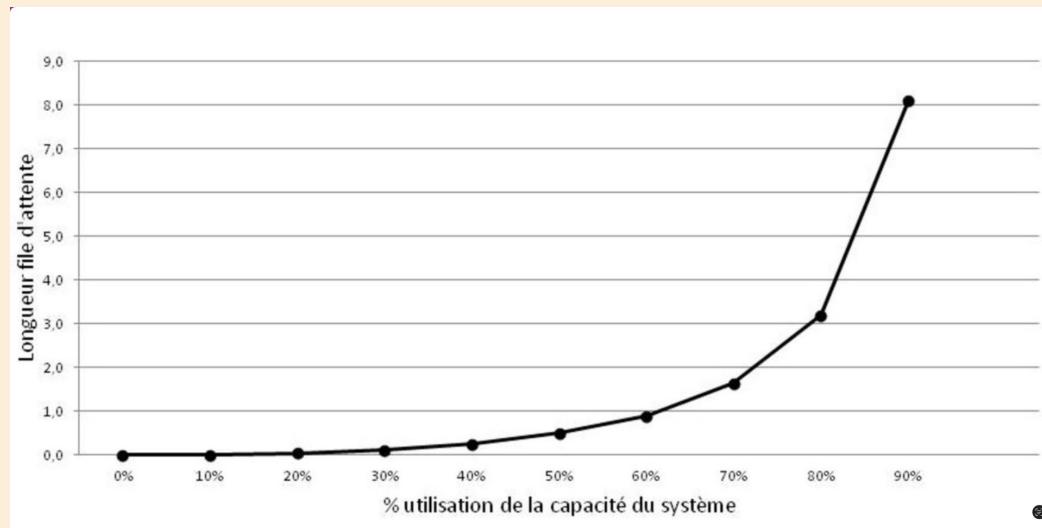
- Gaspillage du Lean
- Diminuer les limites

# Globalement saturé : capacité d'un système

« **Cause** : La demande extérieur excède la capacité de réalisation du système »



## Théorie des files d'attente



La taille de la file d'attente triple si on passe d'une utilisation de 60% de la capacité à 80%

# Globalement saturé : Loi de Little

- Une autre loi est intéressante : Loi de Little (lié à la précédente)
- En Kanban on aura la conséquence suivante

*Si un chef de projet décide d'améliorer l'efficacité individuelle en évitant les blocages ou les attentes par du multitâche (WIP) alors il va produire l'effet inverse. Une augmentation du WIP entraîne une augmentation du délai (temps de cycle)*

Explication en video (10min)

<https://www.youtube.com/watch?v=KO9DRGHRDyc>

# Globalement saturé : Loi de Little

- Le mieux est de l'exprimer sous la forme

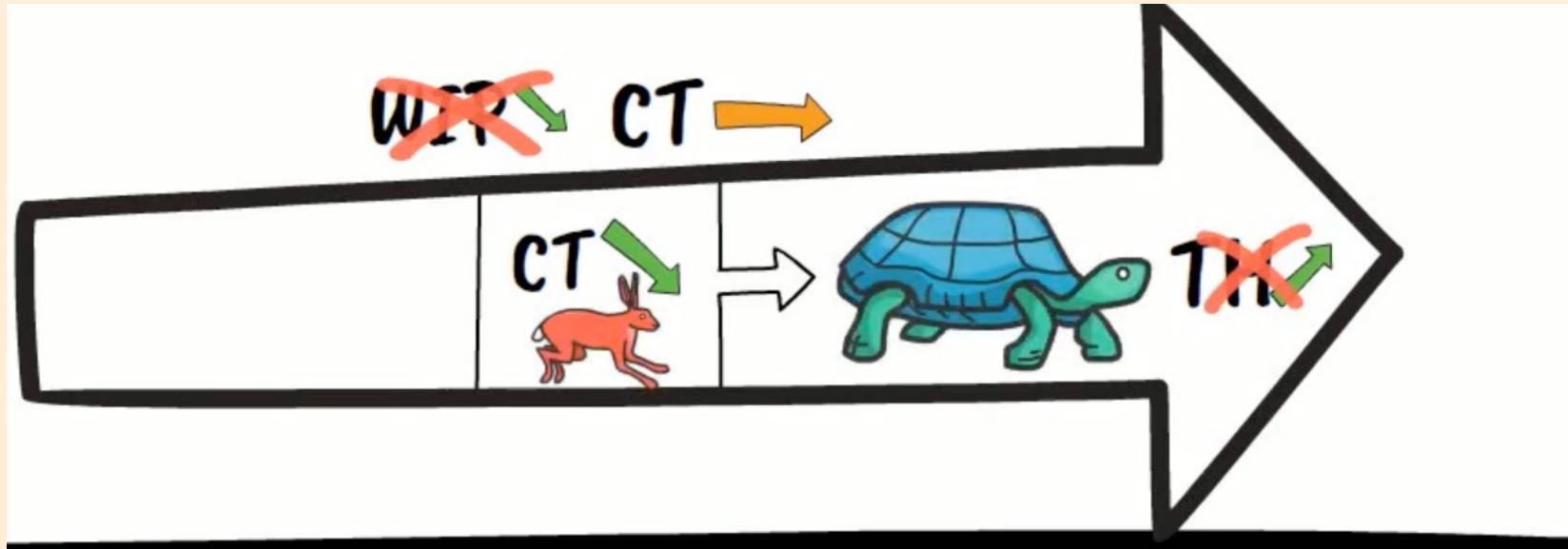
$$\text{CycleTime} = \text{WIP} / \text{Débit}$$

Car c'est le cycle time qu'on ne maîtrise pas du tout

- *Par conséquent*

- Réduire le cycleTime peut se faire en diminuant l'en-cours (WIP)
- Réduire le cycleTime peut se faire en augmentant le débit (e.g. + de devs)

# Globalement saturé : Loi de Little



Si on augmente localement mais qu'en sortie on n'a pas d'effet  
Alors cela signifie que le gain n'est pas absorber par la prochaine équipe

# Globalement saturé : conclusion

- Le meilleur moyen de gérer un système globalement saturé est de réduire la capacité d'absorption du système pour réduire les files d'attente

# Localement saturé

## Étude des dysfonctionnements

### Système globalement saturé

- Théorie des files d'attente
- Ajuster la capacité

### Système localement saturé

- Théorie des contraintes
- Goulet d'étranglement

### Système non saturé mais variable

- Maîtrise statistiques des procédés
- Limites de variabilité

### Limites trop hautes vs capacité

- Gaspillage du Lean
- Diminuer les limites

# Localement saturé

« **Cause** : Localement l'équipe a eu des difficultés à tenir la limite»

# Localement saturé : Théorie des contraintes

## 1. Identifier les contraintes

- Dans le schéma précédent goulot d'étranglement colonne TEST
- Donc empilement des tâches dans FAIT

## 2. Exploiter la contrainte

- **La contrainte fixe la capacité globale du système**
- E.g. Prioriser la contrainte (mettre des experts dessus)

## 3. Subordonner le travail à la contrainte

- Le max de travail que les DEVs doivent faire est au maximum égale à celui des testeurs

## 4. Elever les performances de la contrainte

- Former les testeurs, automatiser le processus ou une partie, etc ...

# Non saturé mais variable

## Étude des dysfonctionnements

### Système globalement saturé

- Théorie des files d'attente
- Ajuster la capacité

### Système localement saturé

- Théorie des contraintes
- Goulet d'étranglement

### Système non saturé mais variable

- Maîtrise statistiques des procédés
- Limites de variabilité

### Limites trop hautes vs capacité

- Gaspillage du Lean
- Diminuer les limites

# Non saturé mais variable

« **Objectif** : Réduire les variations de vitesse, que l'ensemble des éléments avancent au même rythme »



*Il faudrait que toutes le stories aient le même poids ...*

# **Non saturé mais variable**

Mais compliqué, on accepte un minimum de variabilité, figure 17,3 et 17,4