

Kanban

Adrien CAUBEL

Ressources :

- Kanban pour l'IT : Laurant Morisseau

Objectif

Objectif

« L'objectif de Kanban est d'équilibrer la demande client avec la capacité de travail »

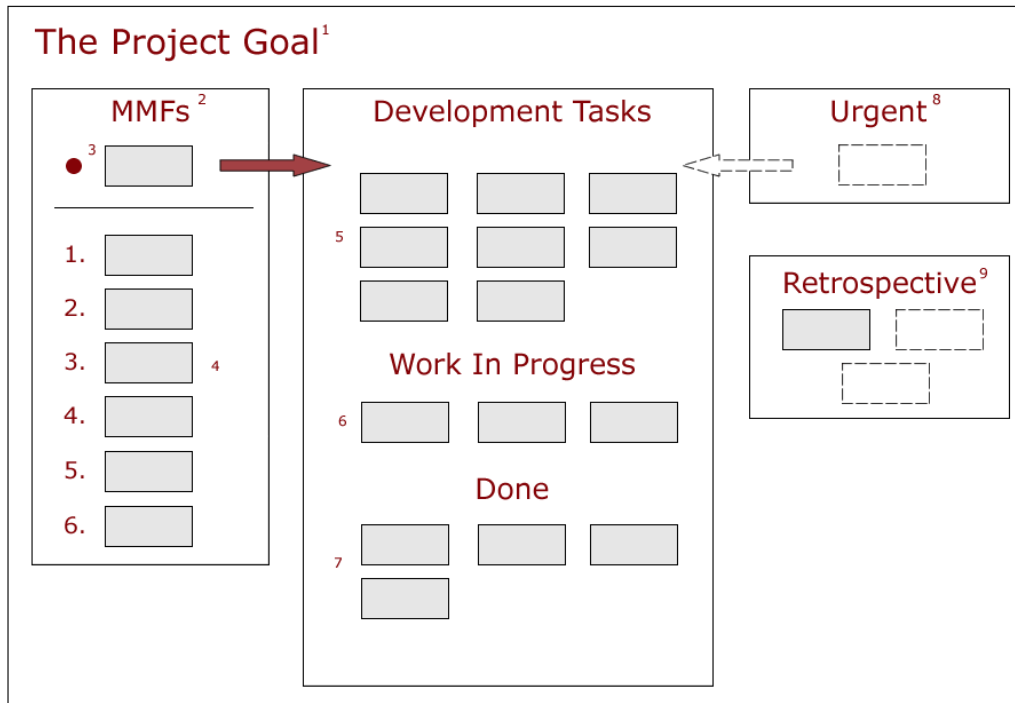
SLIDE IMPORTANTE

« Kanban n'est pas une méthode de gestion de projet ou de cycle de vie de développement logiciel. Il s'agit d'une approche de la gestion du changement - un cadre pour favoriser le changement dans une organisation »

Scrum, Kanban, etc ... on vise l'amélioration de l'organisation

« Kanban n'est pas une méthode de gestion de projet ou de cycle de vie de développement logiciel. Il s'agit d'une approche de la gestion du changement - un cadre pour favoriser le changement dans une organisation »

Kanban Board



Gérer le flux de A à Z

Et pas qu'optimiser le temps de développement

1 - The overall goal of the project and if applicable the objective of the current release cycle.

2 - **Minimal Marketable Feature** - The smallest feature that provides value to your customer.

3 - The current MMF that the development team is working on.

4 - The backlog of MMFs. The order of these can be changed at any time by the project's stakeholders.

5 - The development tasks that are required to satisfy the current MMF.

6 - The development tasks currently being worked on.

7 - Development tasks for the current MMF that have been completed.

8 - An urgent need that skips the backlog.

9 - A queue for completed MMFs that triggers a retrospective once full.

Définition

Définition

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

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.

Comment

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

5 propriétés maitresses

1. Visualiser le flux de travail
2. Limiter les travaux en cours (WIP)
3. Gérer et mesurer le flux de travail
4. Expliciter les politiques de processus
5. Améliorer en collaboration

« Utilisation » de Kanban

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



La méthode Kanban nous dit uniquement de gérer le *flow of work*
et le *work in progress*



Pas de périmètre fixe

Management visuel

(On y revient plus tard en détail)

Pourquoi l'utilisé

- 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.

La tableau

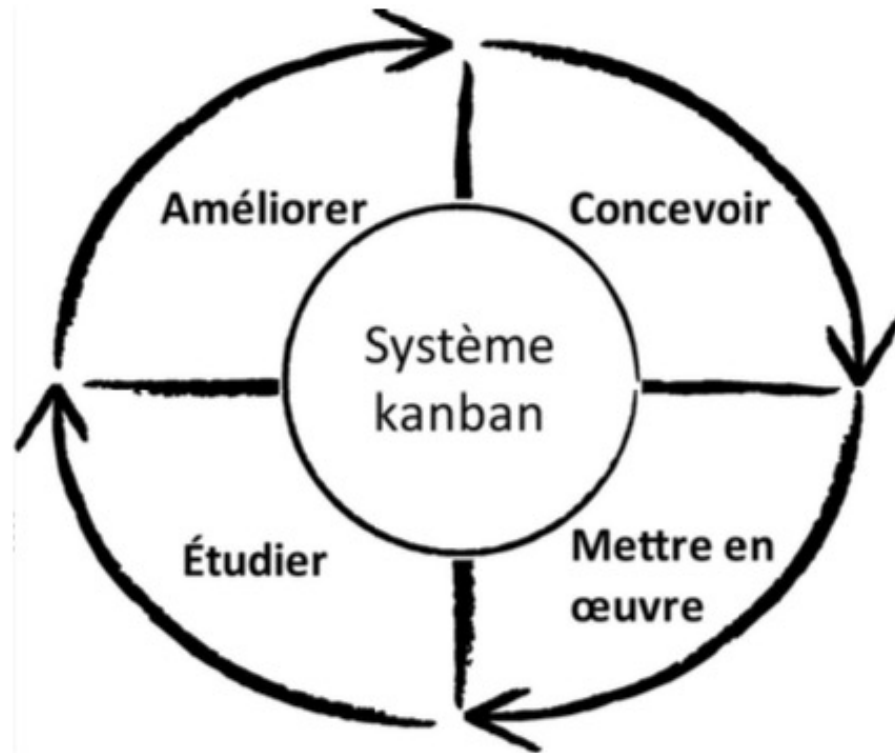
- Il doit être compréhensible par tous sans explication ;
- Il doit respecter un certain nombre de codes visuels ;
- Il doit être lisible à quelques mètres de distance pour la réunion quotidienne ;
- Il doit être facile à manipuler par tous, c'est l'outil de l'équipe ;
- Il doit être facile à faire évoluer ;
- Il doit être facilement accessible, si possible dans l'open space de l'équipe.

Visualiser via des cartes

- La date d'entrée dans le système ;
- La date de sortie réelle pour calculer a posteriori, avec la date d'entrée, le temps de traversée
- Les informations pour effectuer le travail : résumé d'une fonctionnalité, libellé d'une story...
- Le type de travail

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

Le concepteur du système Kanban doit :

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

Est réalisé avec l'équipe et les représentants des parties prenantes

2. Mise en oeuvre

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

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

3. Etudier

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

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

4. Améliorer

- L'équipe tire les enseignement 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

Concevoir



5 pratiques

3 piliers des fondations

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

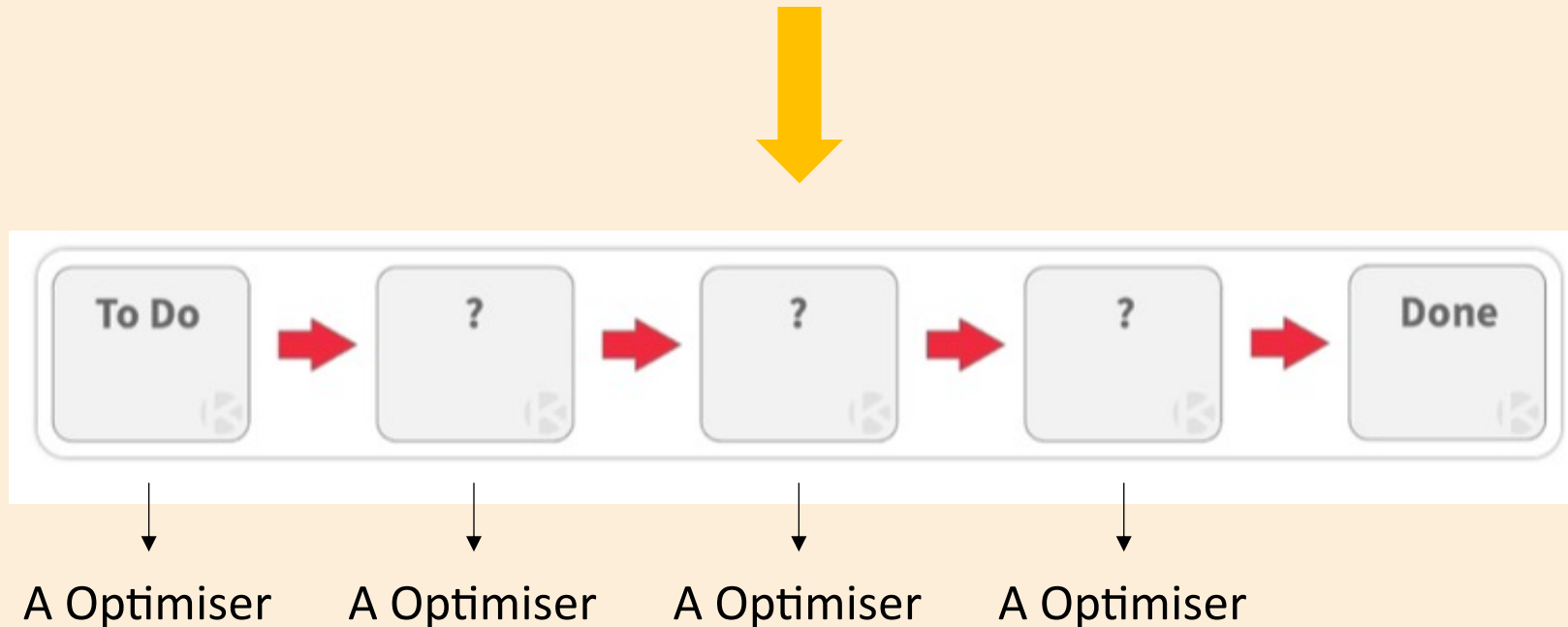
Cadence

Nature de la demande



Objectif

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



Élément 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 »

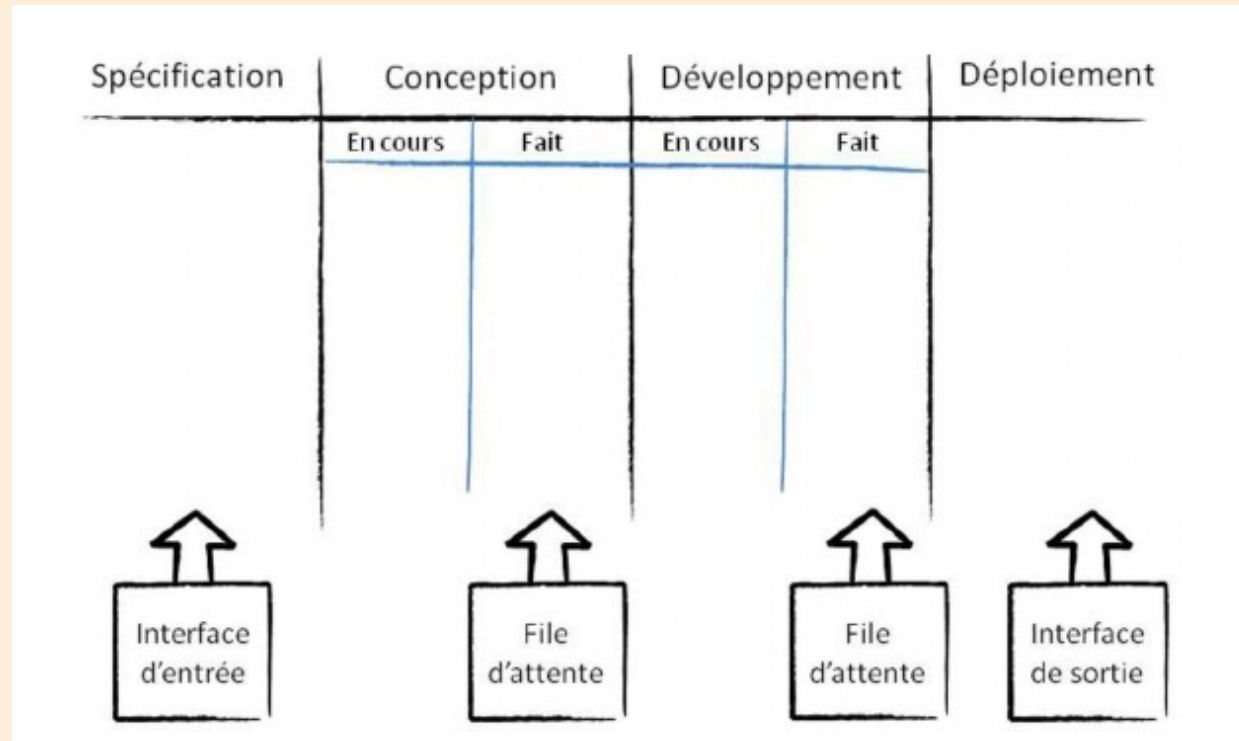
Élément de travail

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

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



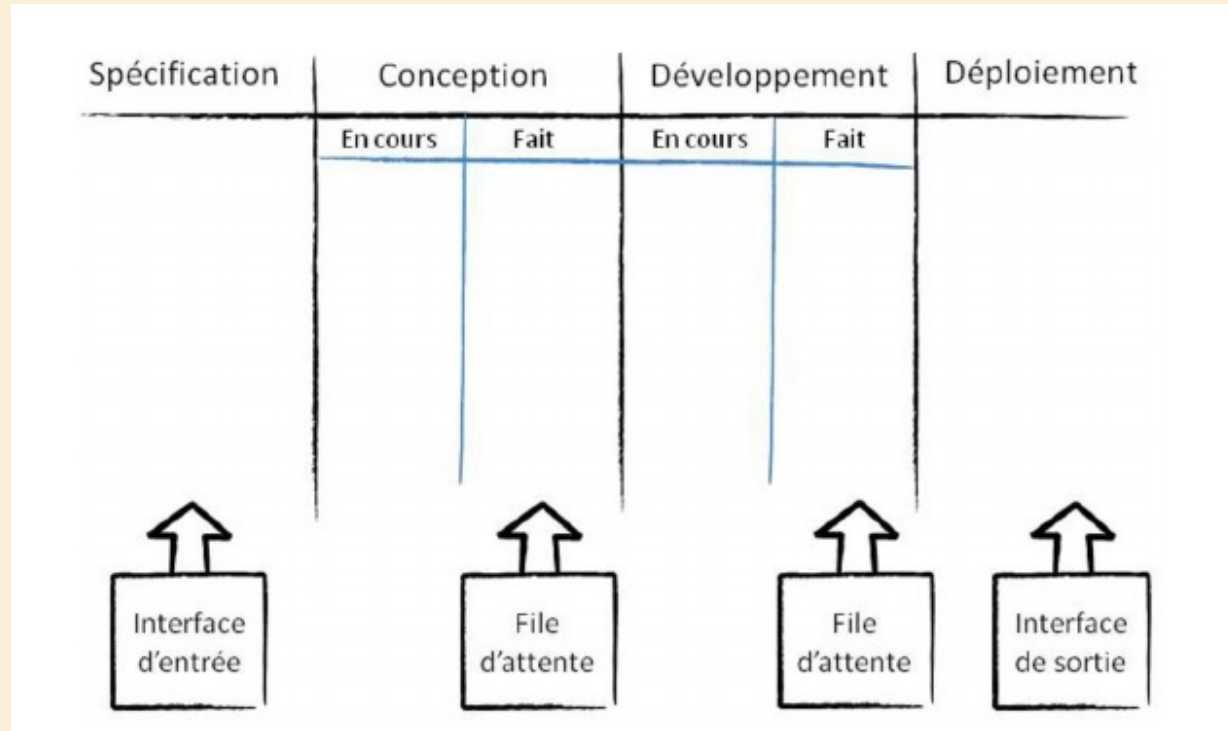
Visualiser le flux de travail



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) sont prêts (pour la conception ici) mais qui ne peuvent être encore *en cours* car il n'y a pas de ressource disponible.

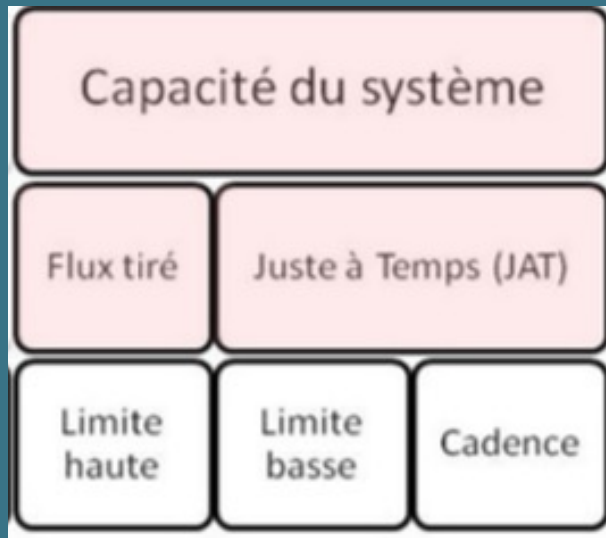
File d'attente : sépare les équipes spécialisées pour leur permettre de travailler de manière asynchrone.

Des kanban dans Scrum ?!

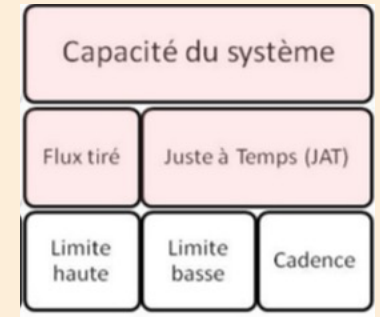
- On voit souvent des kanbans dans Scrum
- Parce que la théorie Scrum valorise la transparence et la visibilité pour faciliter le cycle d'inspection et d'adaptation.
- Les tableaux aident à visualiser le travail, et cette pratique est donc couramment empruntée à Kanban.

⇒ **Le tableau est juste une aide visuelle.**

Capacité du système

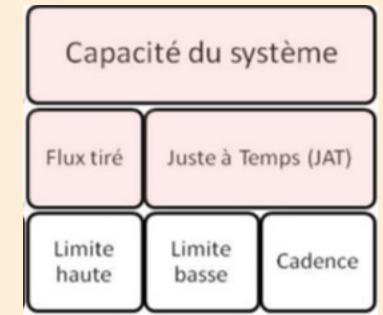


Flux tiré



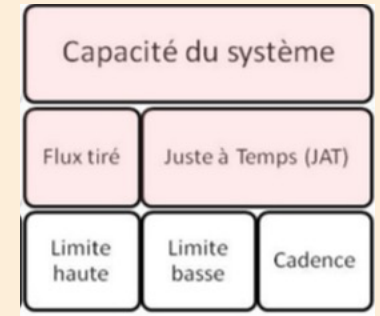
« Dans un système "pull » (tiré) une tâche est récupérée uniquement lorsque l'équipe a la capacité de la gérer. »

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 vélocité 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)

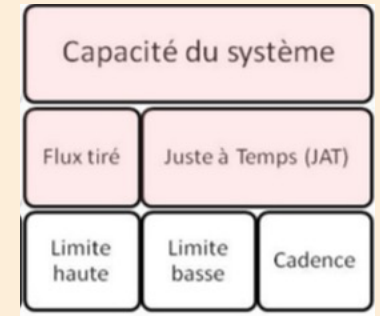


« Le *Work in progress* (limite de travail) définit la quantité maximale de travail qui peut exister dans chaque colonne. »

Le WIP permet de :

- 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)

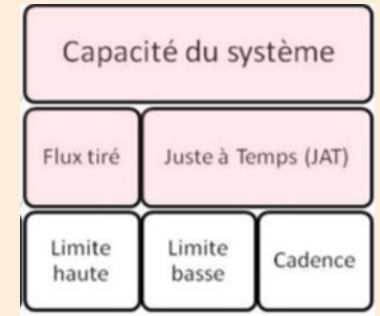


Le WIP d'éviter :

- Sur-engagement : on a une capacité max on peut pas s'engager au delà
- Multitâche
- Sur-production : capacité maximum de livraison est définie
- Amélioration continue : POURQUOI ??

les limites sont des contraintes qui mettent le système sous tension => provoque le changement

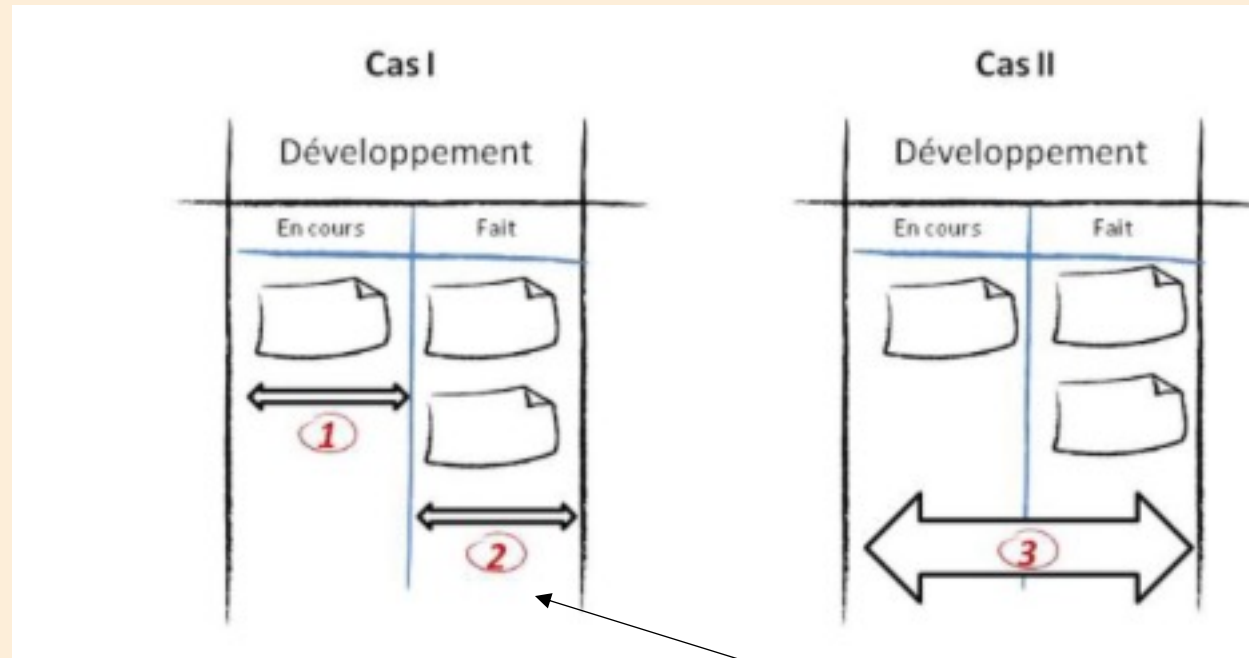
Flux tiré : WIP par type (optionnel)



Si WIP = 10

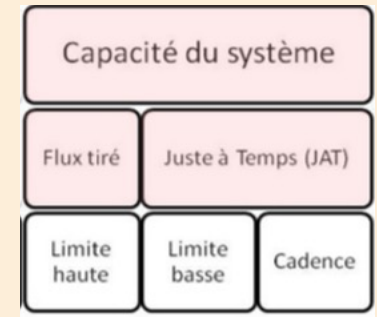
- Anomalie : 10% => max 1 ano dans la colonne
- Evolution : 70% => max 7 evol dans la colonne
- Review : 20% => max 2 evol dans la colonne

Flux tiré : WIP précis

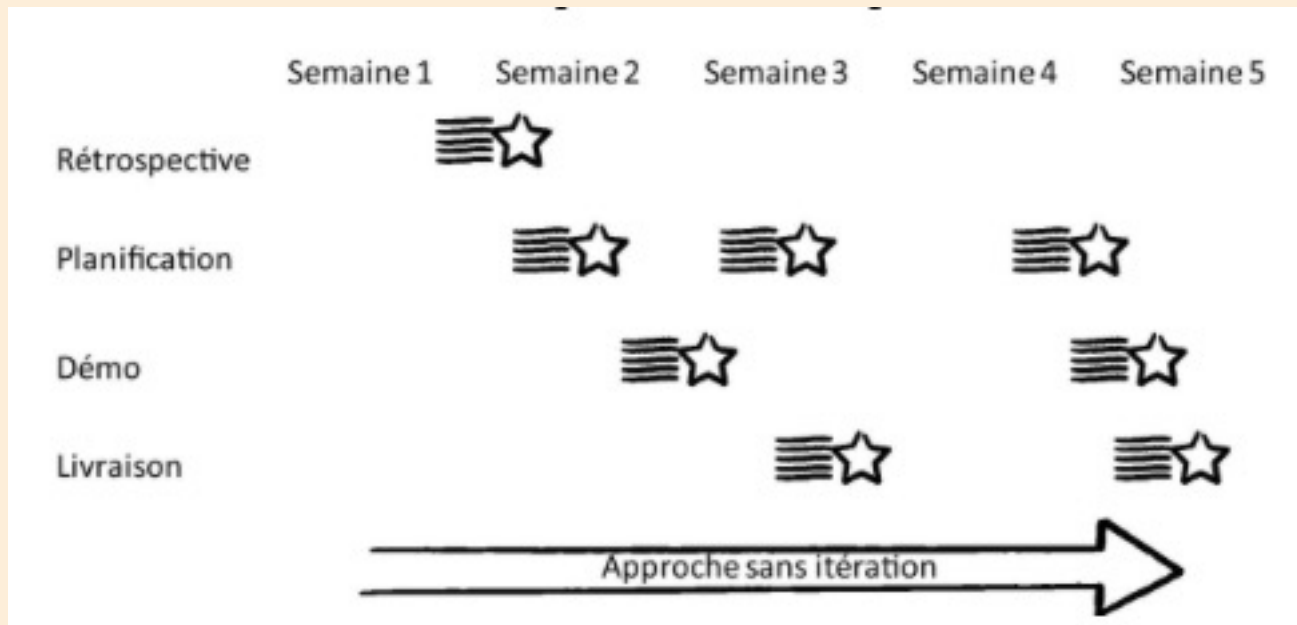


Proposition : Mettre une WIP sur une file d'attente

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 »

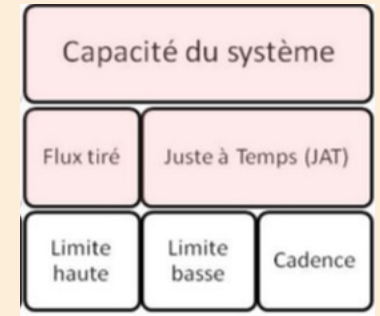


- Repousser la prise de décision
- Traiter à la demande
 - Injection d'élément de travail à la demande
 - Livraison dès que release prête

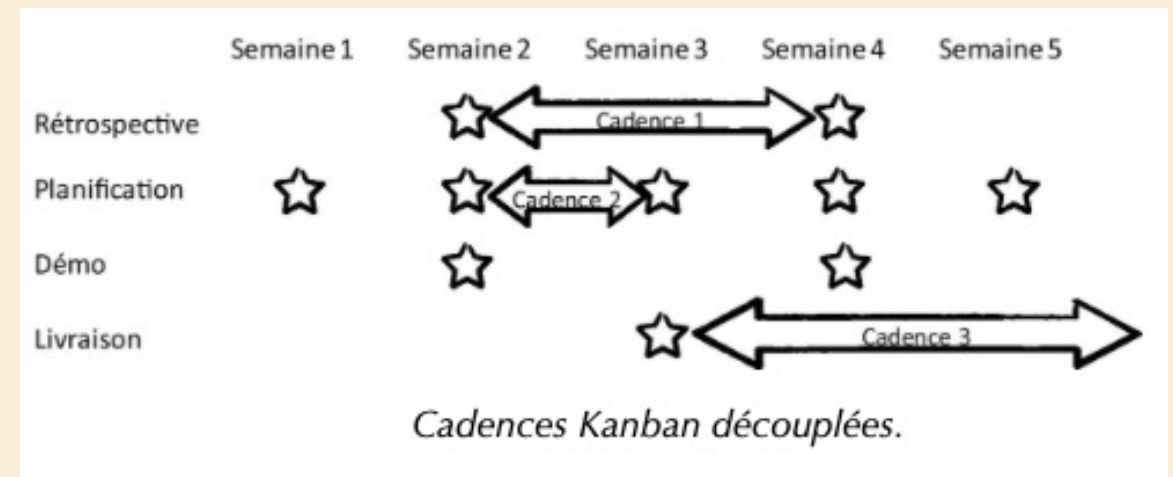
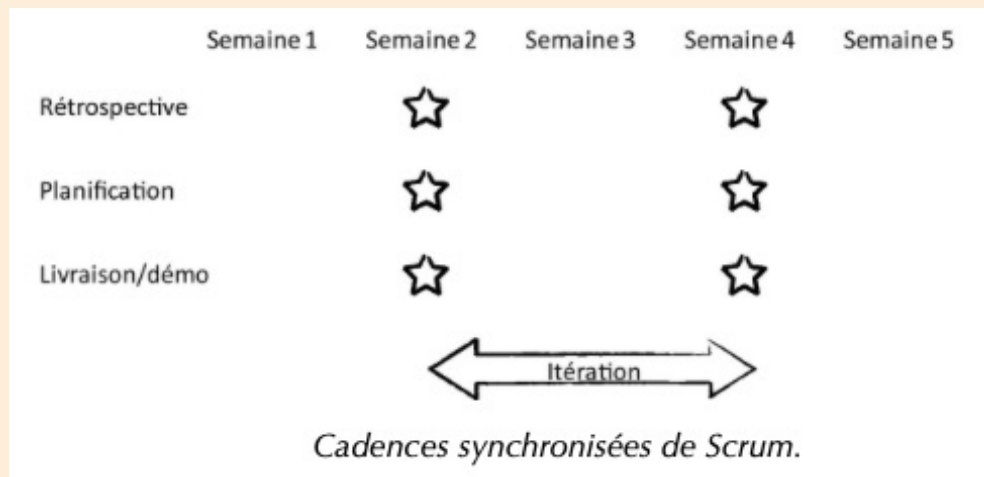
JAT sert à réduire le temps entre une prise de décision et l'effet qu'elle va produire

Pas besoin d'attendre n semaine avant une planif : JAT

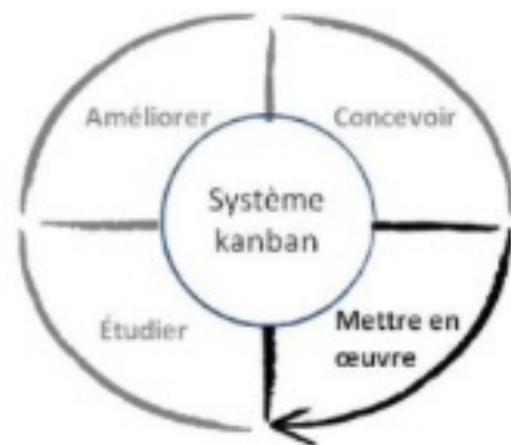
Juste à temps : cadence



« Chaque activité peut avoir un cadence différente des autres »



Mise en oeuvre



5 pratiques

3 piliers des fondations

Système kanban au quotidien

Réunion quotidienne

Mouvement
d'un élément

Suivi du
système

Sélection

Anomalie

Blocage

Carte de
contrôle

Débit

CFD

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
 - 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 SG)
 - Ici être stratégique : choisir le « bon » type pour ne pas bloquer le système, le plus rapide à faire et le plus prioritaire

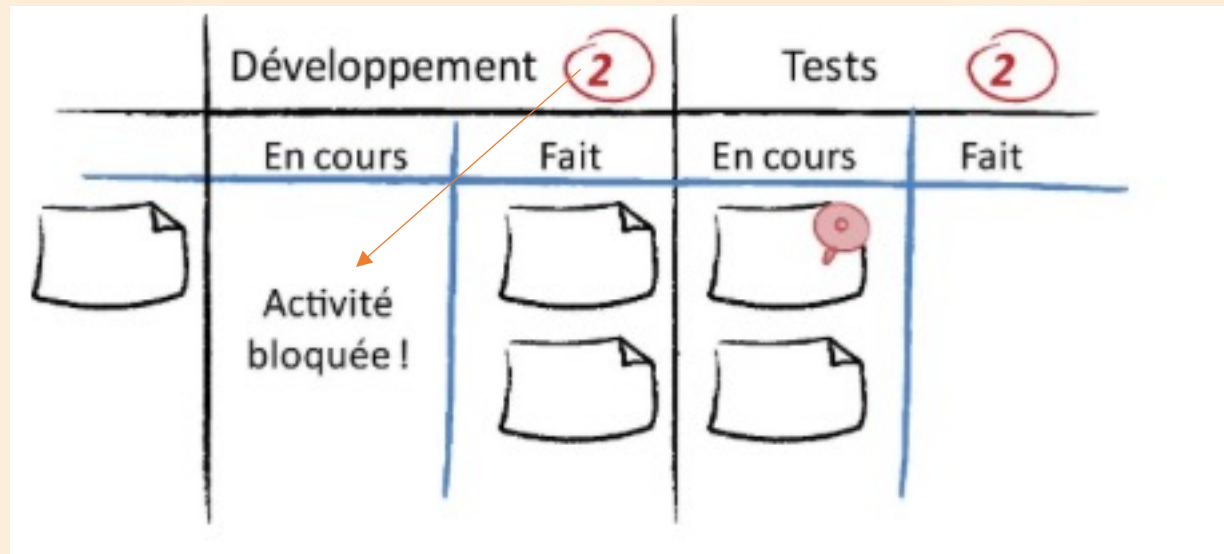
Gérer l'affectation

- Affectation faite au dernier moment
- Affecter trop tôt contribue à l'accumulation du travail (stock => lean)
- L'affectation 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



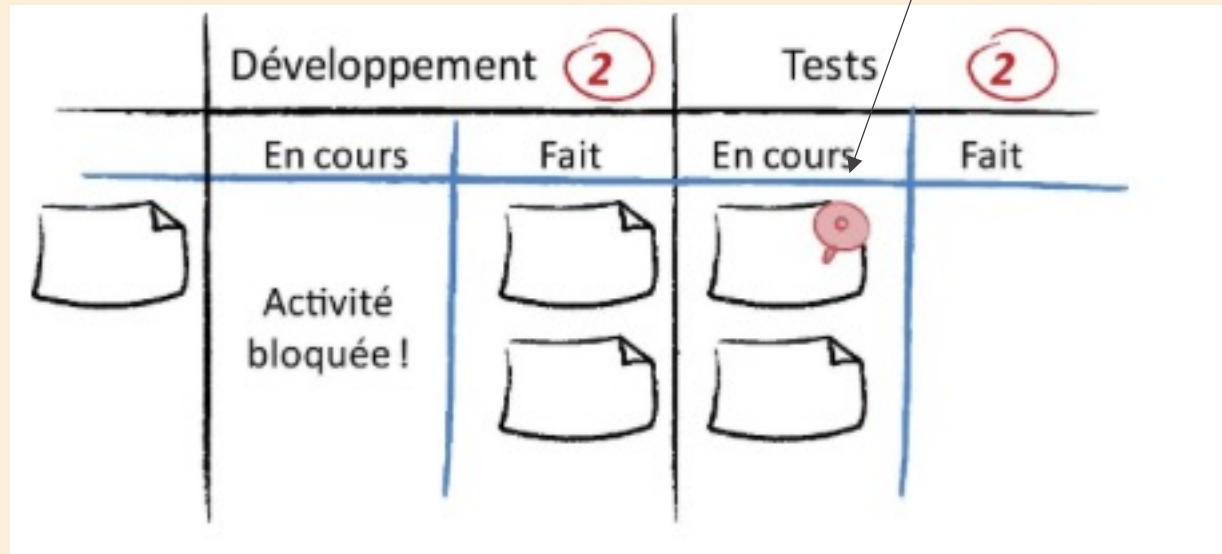
Les devs attendent que le système se débloque ??

Gérer le blocage : répondre par la demande

- Ne pas commencer un travail si on sait qu'il va être bloqué
- Débloquent 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



Je vais dans l'équipe de tests

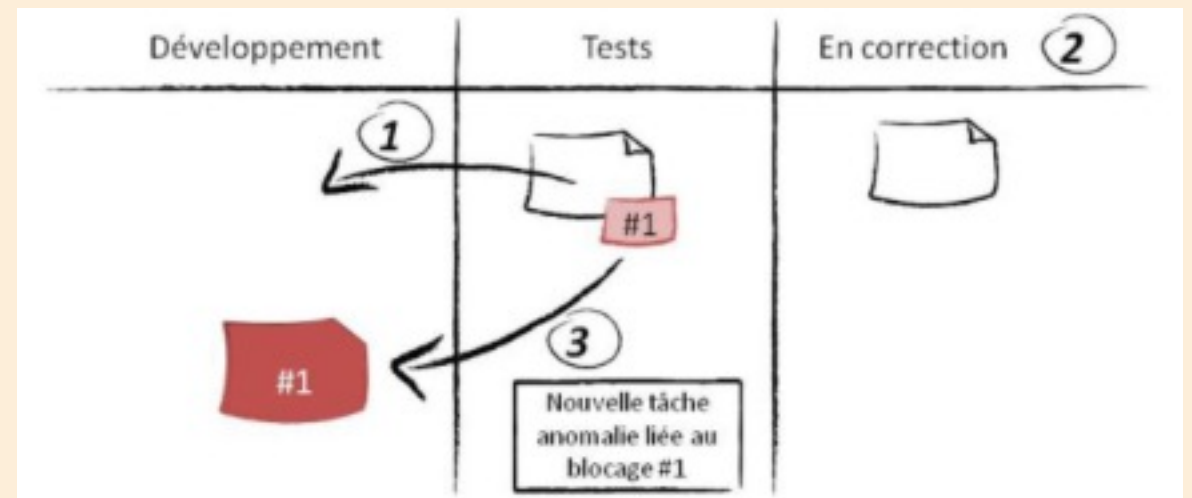
Gérer le blocage : répondre par l'ajustement des règles

- Ni la gestion de la demande
- Ni la capacité débloque le système
- => Ajuster la règle (démarche empirique)
 - Augmenter la WIP

Gérer les anomalies

- Que faire si une carte kanban se retrouve bloquée ?

1. Est-ce qu'elle revient en arrière ?
2. On crée une colonne en correction ?
3. Créer une nouvelle tâche : anomalie (permet de tracer)



Suivi du système



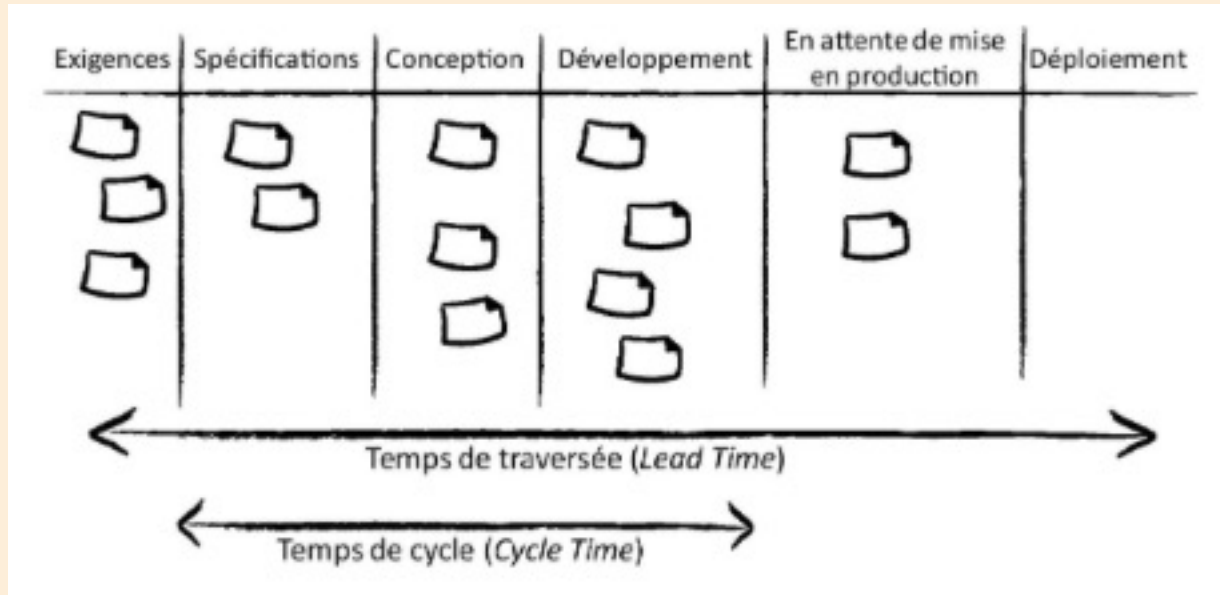
Mettre à jour les indicateurs



« Lors des réunions quotidiennes 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



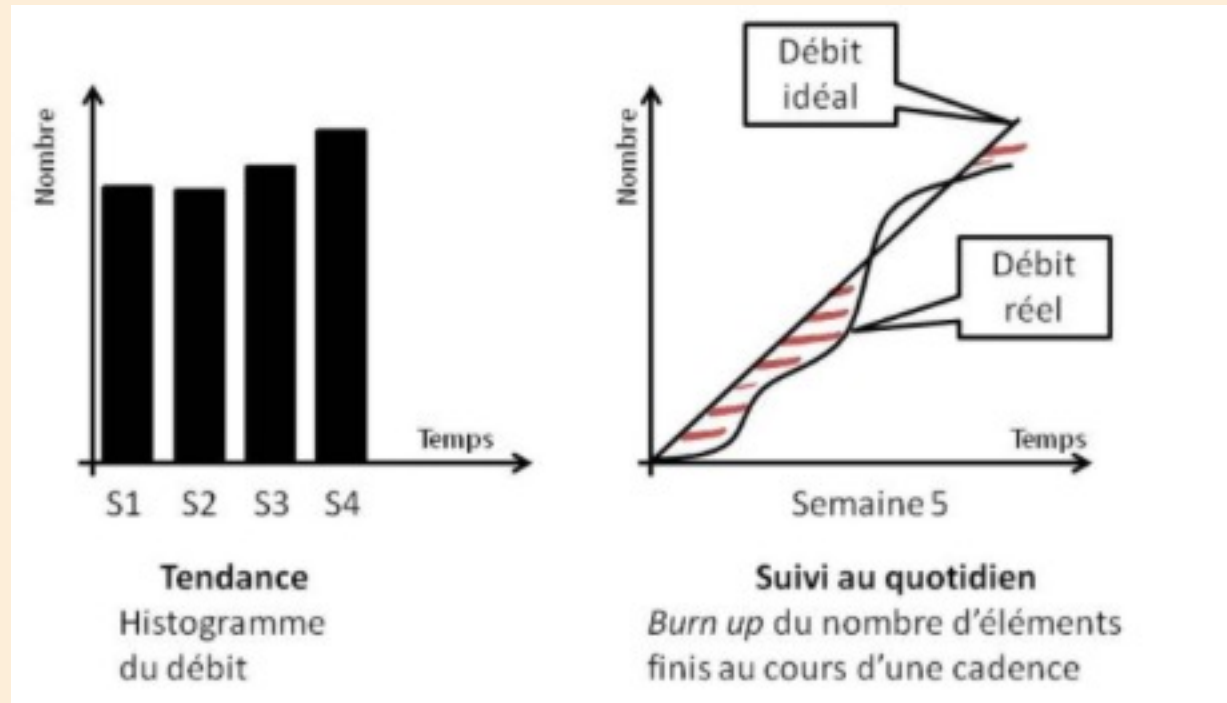
Débit correspond :

« *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)

Suivre le débit : burnup chart



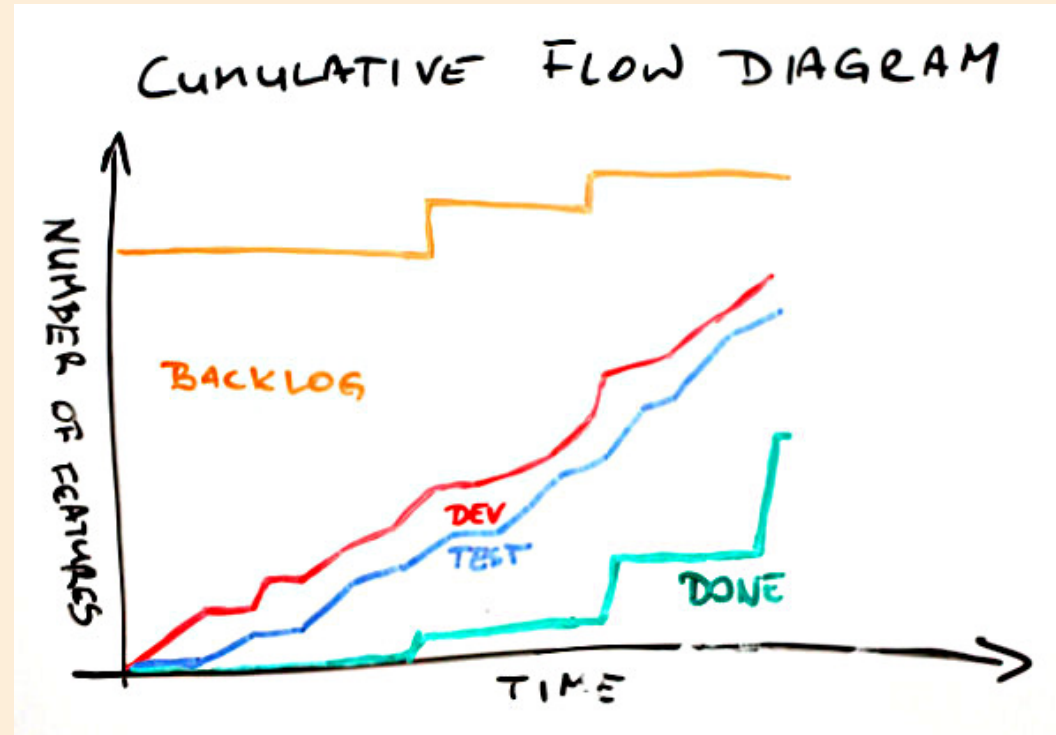
On représente le nombre d'éléments finis (SP) en fonction du temps

Puis on compare la courbe réelle à la courbe idéale

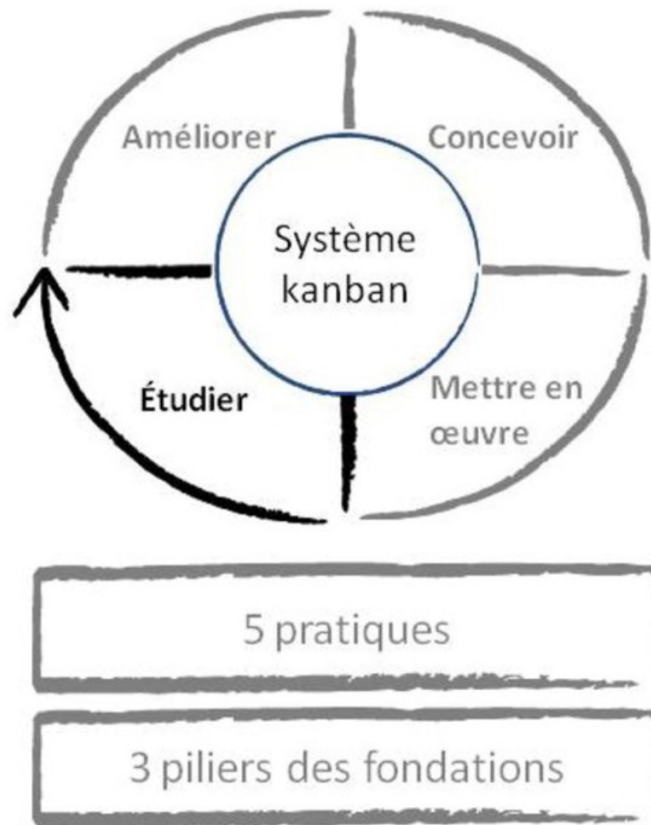
Suivre le nombre d'éléments dans le système



Suivre le nombre d'élément en cours par activité



Etudier un système Kanban



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

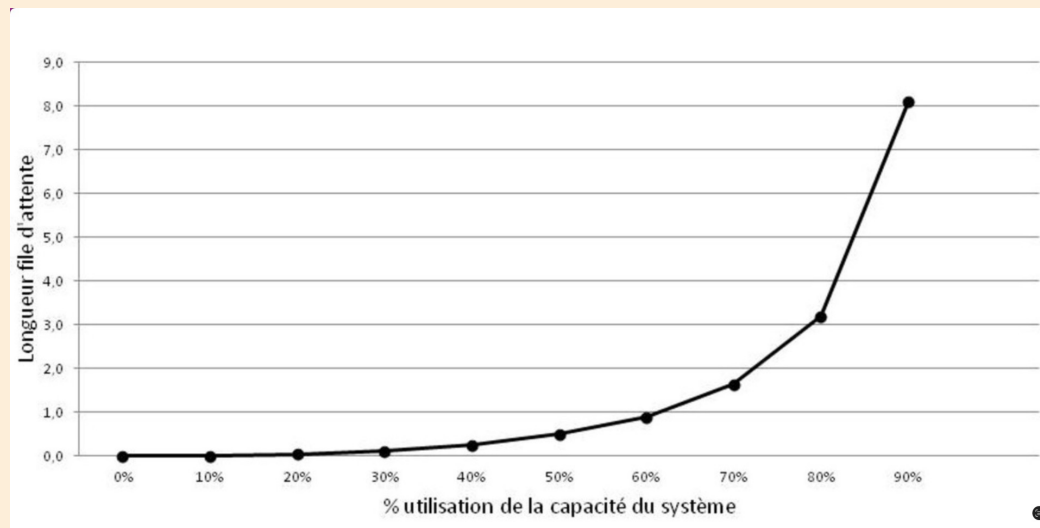
- Gaspillage du Lean
- Diminuer les limites

Globalement saturé : capacité d'un système

« Cause : La demande extérieure 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 délais (temps de cycle)

Explication en video (10min)

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

Globalement saturé : Loi de Little

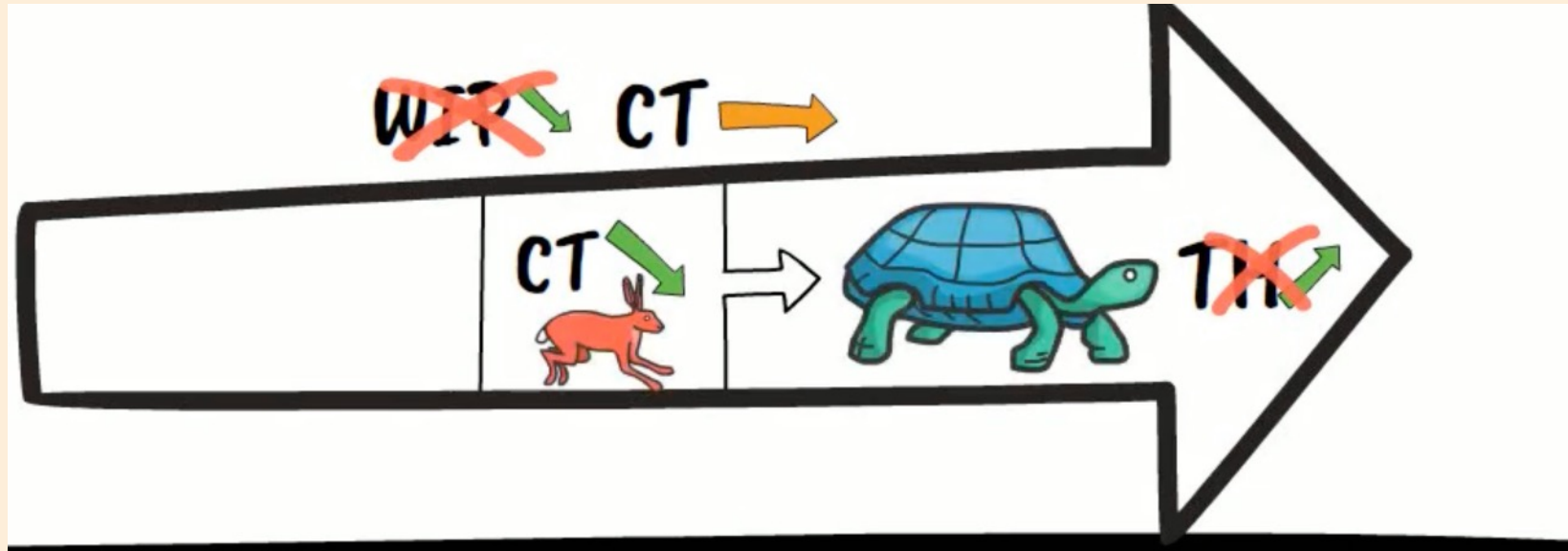
- Le mieux est de l'exprimer sous la forme

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

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 absorbé 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