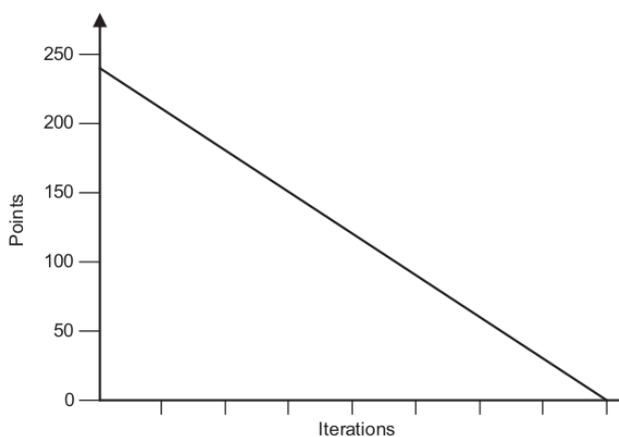


Definition. *Le burndown chart s'adresse à l'équipe. Il représente le reste à faire*

Affirmation 1. *Si la ligne de travail réelle est supérieure à la ligne de travail idéale, cela signifie qu'il reste plus de travail que prévu et que le projet est en retard.*

Affirmation 2. *Si la ligne de travail réelle est inférieure à la ligne de travail idéale, cela signifie qu'il reste moins de travail que prévu et que le projet est en avance sur le calendrier.*

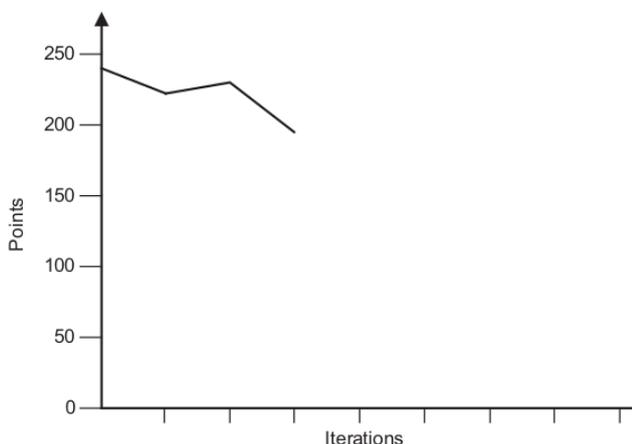
1 Release burndown chart



- X : le nombre d'itérations pour finir la release
- Y : Le nombre de *story-points* (SP)

Le diagramme ci-contre montre un burndown chart idéal avec une release de 240 story-points délivrées de manière égale en 8 itérations. Mais il ne faut pas s'attendre à ce que l'équipe est une *vélocité* de 30 tout au long de la release.

FIGURE 1 – Ligne de travail idéale



Voici un diagramme plus probable après 3 itérations :

- Itération #1 : 30 SP accomplies
- Itération #2 : Augmentation
 - L'équipe se rend compte qu'une story est plus compliquée. Elle augmente donc sa valeur
 - Du travail a été ajouté; un oubli ou quelque chose qui apporte vraiment de la valeur a été découvert

FIGURE 2 – Ligne de travail réelle

Definition. *Le burndown chart nous montre si une release est sur la bonne voie :*

- *Le taux d'avancement de la release*
- *La quantité de travail restante*

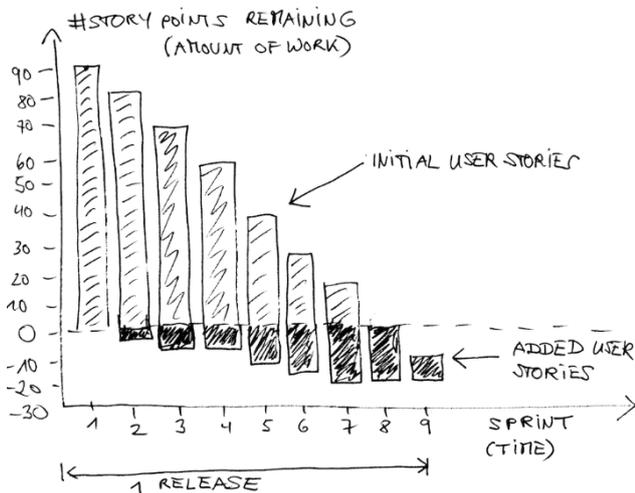
Nous dit donc quand la release devrait se finir

À la fin de la troisième itération, on va se demander *si la release sera finie au bout de 8 itérations* et par conséquent ajuster l'estimation.

2 Alternative Release burndown chart

Definition. Permet d'observer la variation de la vélocité ET la variation du périmètre séparément.

En effet, dans le graphique de la section précédente on ne sait pas pourquoi il y a eu une augmentation des SP (SP ajoutés ou SP mal estimés ?). Ici si des SP ont été ajoutés alors augmentation sous l'axe des abscisses. Et si des SP mal estimés alors augmentation au-dessus de l'axe des abscisses (ou diminution moins rapide == vélocité réduite)



— X : Le nombre d'itération

— Y : Le travail restant pour finir la release^a

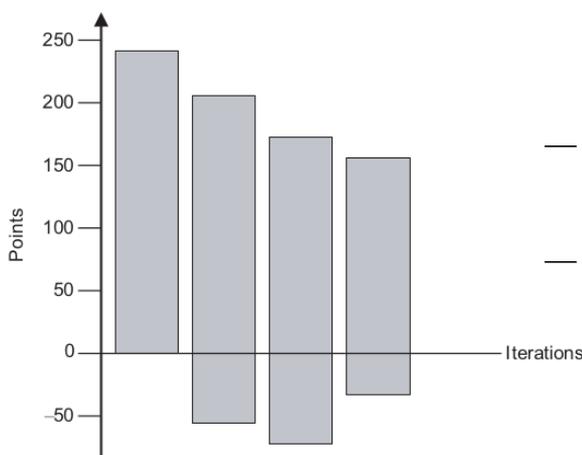
Lorsqu'on rajoute du travail (SP) dans un sprint on l'ajoute sous l'axe 0

Le Product Owner met à jour le Release Burn Down à chaque Sprint en se basant sur la vélocité et les estimations du développeur. Souvent, le Release Burn Down est utilisé pour l'ensemble du produit, montrant le Burn Down pour toutes les versions.

a. version

2.1 Lecture de l' alternative release burndown chart

Chaque barre montre la quantité de travail dans la release au début de l'itération (préfixe d_).

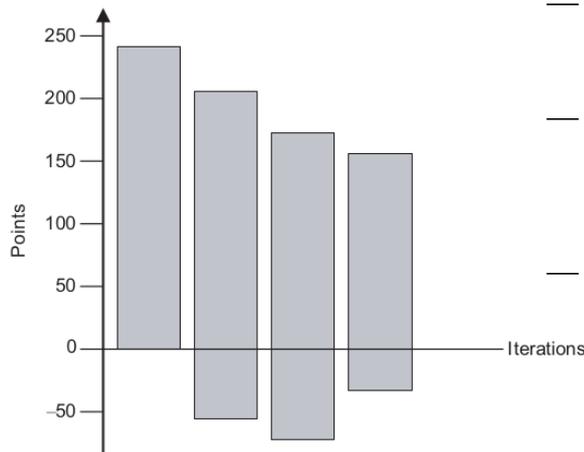


— d_Itération #1 : la release nécessitait 240 SP et la vélocité de l'équipe est de 30 (on la connaît)

— d_Itération #2 : durant I1 vélocité OK ⇒ reste 210 SP
MAIS le PO en a identifié 50 nouveaux.
⇒ La release est de maintenant 260 SP.

À la fin de la 2ème itération, le graphique nous dit 3 choses :

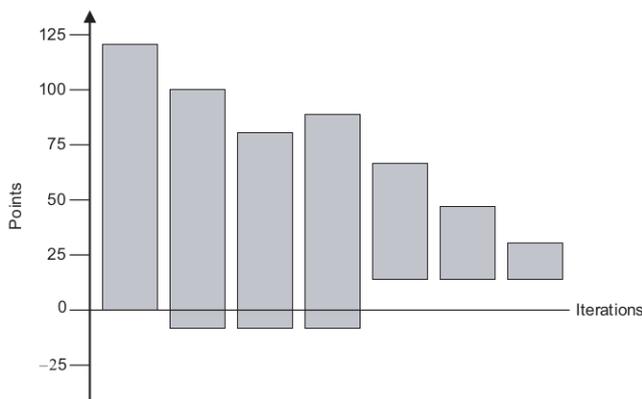
1. La vélocité de l'équipe est conforme aux attentes. A chaque itération 30 SP ont été achevés.
2. Du travail a été ajouté. On peut supposer qu'il a été ajouté, car il apporte une plus grande valeur. Il faudra faire attention, car il y a plus de travail ajouté qu'achevé : est-ce que la date de livraison a été adaptée ?
3. La quantité de travail restante est plus importante qu'au début du projet (hauteur de la deuxième barre supérieure à la première)



- d_Itération #3 : durant I2 l'équipe a maintenu la vélocité \Rightarrow 180 SP MAIS PO à identifier +10SP.
- d_Itération #4 : durant I3 vélocité a diminué, passant de 30 à 20
 - Personnel absents
 - US plus dures
- d_Itération #4 (bis) : durant I3 le PO a supprimé des items
 - Mais toujours sous l'axe 0 donc toujours plus de SP qu'initialement prévus
 - Aucune importance si le feature été initialement prévu où a été ajoutée durant une itération

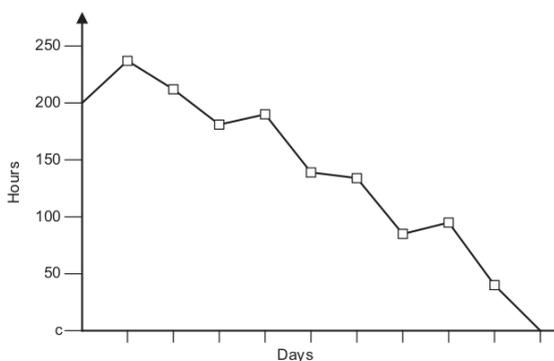
Quatre règles

- Chaque fois que le travail est achevé, le sommet est abaissé.
- Lorsque le travail est réestimé, le sommet est relevé ou abaissé.
- Lorsque de nouveaux travaux sont ajoutés, le bas est abaissé.
- Lorsque des travaux sont supprimés, la base est relevée.



- Itération #3 : L'équipe se rend compte que des US ont été mal estimées. Elle ré-ajuste leur SP. Donc augmentation sur la 4ème barre
- Itération #4 : au début de l'itération le PO supprime du travail. Puis l'équipe va reprendre un rythme normal.

3 Iteration/Sprint Burndown Chart



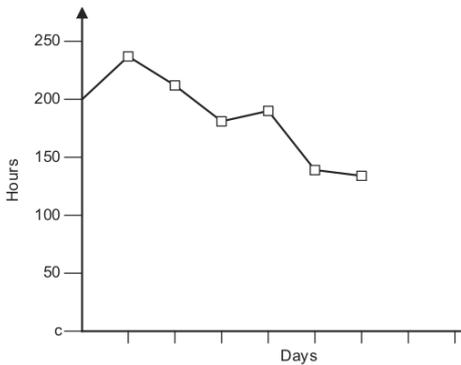
Quand on travaille par sprint et non en release on peut adapter notre Burndown Chart. On peut également utiliser des SP au lieu des heures

- X : Le nombre de jours dans l'itération
- Y : Le nombre d'heures restantes à faire dans le sprint

Affirmation 3. Si la courbe de travail réelle augmente où que les SP sont ajoutés sur l'axe 0 Alors l'équipe a estimé qu'une tâche est plus dure qu'imaginée.

Ceci n'est pas très grave à condition que ça ne se répète pas trop souvent

3.1 Vitesse de l'équipe : OUI!



Presque à la fin du sprint et encore plus de la moitié des SP à tomber.

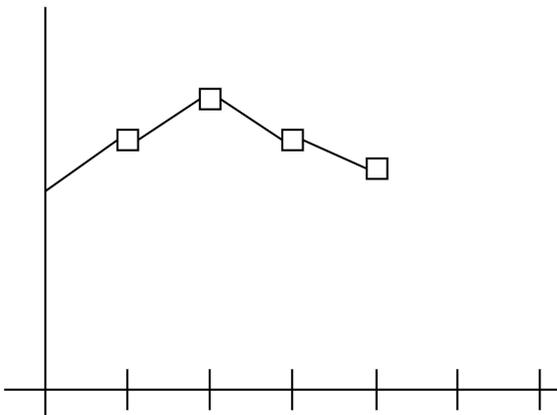
- L'équipe a peut-être commencé des US mais ne les a pas finies. Le Scrum Master peut regarder le *Cumulative flow diagram*
- Le poids des US n'a pas été significativement augmenté. Donc pas de problèmes techniques majeurs

3.2 Vitesse individuelle : NON!

Certaines équipes se réfèrent à la vitesse individuelle comme étant le nombre de points d'histoire ou de jours idéaux réalisés par un membre de l'équipe. Le suivi de la vitesse individuelle conduit à un comportement qui va à l'encontre de la réussite du projet. Si je suis obligé de choisir entre terminer une US et aider quelqu'un d'autre, quelle incitation la vitesse individuelle va me donner ?

4 Les dérivés du burndown chart

4.1 Le mauvais chef de projet



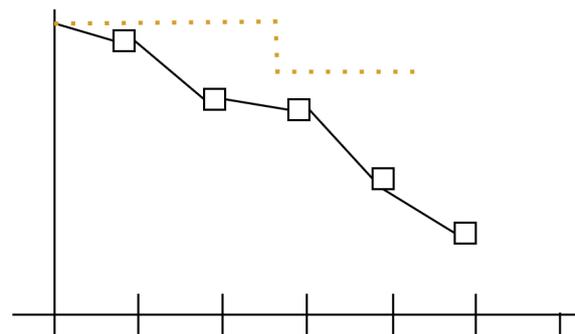
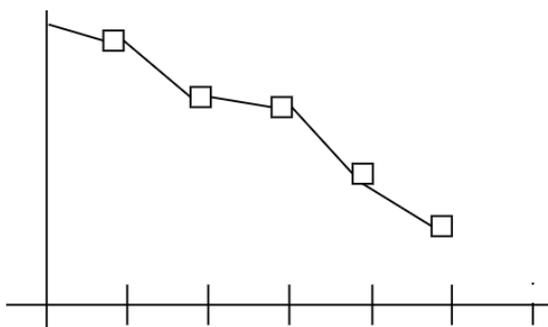
Il regarde le burndown-chart et voit que ça n'avance pas

- Il est stressé
- Il propose à l'équipe de rester plus tard les prochains soirs et paye les pizzas.

Le burndown-chart **est pour l'équipe** et là il ne l'aide pas

- Arrêter ou cacher le burndown-chart

4.2 On pense avoir bien avancer



Sur le graphique de gauche, il semblerait que l'équipe est super bien avancée et va finir la fonctionnalité à la fin du sprint. Toutes les tâches techniques sont presque finies, si on compte en heure, la tâche fait 5h il reste 1h de travail. Donc la courbe est parfaite, il reste quelques heures sur chaque story MAIS aucune story n'est finie.

Ainsi à droite on représente les tâches/heures restantes ET les stories finies. Car au final on souhaite maximiser le nombre de stories finies (DONE).

Affirmation 4. *Soit c'est fini soit ce n'est pas fini*

5 Conclusion

Le burndown chart est pour l'équipe.

Il existe différents types de burndown chart (axeY : SP ou heures - axeX : itérations ou jours). Les équipes agiles préféreront peut-être travailler avec le *Spring Burndown Chart* qui représente l'avance du sprint. Tandis que le chef de projet travaillera avec le *Release Burndown Chart* représentant l'avance d'une release à travers plusieurs itérations.