

# Faire du ScrumBan avec iceScrum

iceScrum a été commencé il y a longtemps, avant qu'on parle de Kanban dans les méthodes agiles. On parlait à peine de Scrum à l'époque. Mais lors de la migration en Grails en 2010, nous avons en tête des pratiques Kanban avec l'idée d'en intégrer progressivement.

Avec la version actuelle, pas de vrai Kanban dans iceScrum, mais il est possible de faire du ScrumBan, c'est à dire de mixer des pratiques de Scrum et de Kanban.

Il existe de nombreuses possibilités pour prendre le meilleur des 2 (pour reprendre le titre de la traduction que nous avons faite du mini-livre de Kniberg et Skarin, voir <http://www.aubryconseil.com/post/2eme-version-francaise-de-Kanban-et-Scrum>). Je vais en décrire une, très orientée kanban, adaptée pour des petites équipes distribuées, voire pour une seule personne qui n'a pas de mur pour des post-it, et permettant de gérer un flux des travaux.

Dans cette optique, on ne va utiliser qu'un sous-ensemble d'iceScrum : le plan de sprint (qu'on pourrait rebaptiser tableau kanban) et le bac à sable. On n'utilise pas le backlog, et quasiment pas le plan de release.

## Contexte

La technique que je vais présenter peut se mettre en oeuvre sur un nouveau projet, ou sur une nouvelle release, voire un nouveau sprint d'un projet en cours.

Pour cette utilisation d'iceScrum il n'est pas besoin de changer les options dans les pratiques du projet, sauf pour les tâches urgentes, qu'il faut demander d'afficher. Il convient de fixer une limite pour ces tâches dans l'état en cours. Pour commencer on peut par exemple, prendre 2 fois le nombre de personnes dans l'équipe.

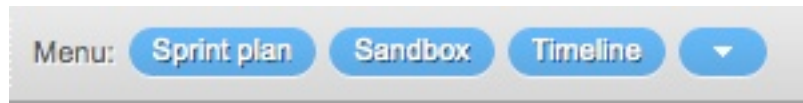
Practices

- ▶ ...for planning
- ▶ ...for sprint management
- ▼ ...for urgent tasks management
  - Display urgent tasks Yes  No
  - Limit of urgent tasks in progress (0 = no limit)
- ▶ ...for meeting hours management

Update Cancel

## Configuration du menu

Puisqu'on utilise seulement quelques vues, autant masquer les autres. Cela se fait simplement en faisant glisser les boutons.



Nous allons essentiellement utiliser la vue Plan de sprint (on pourrait presque n'utiliser que ça), alors mettons là à gauche.

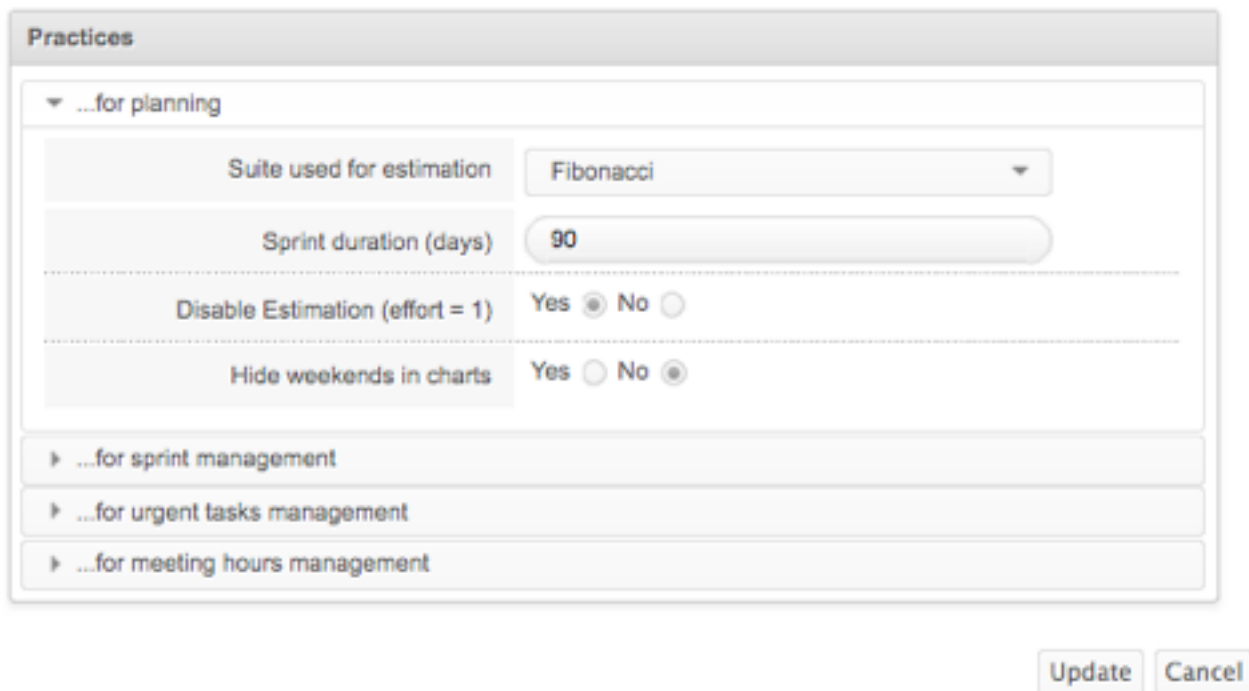
## Sprint ?

Dans une approche orientée flux, les sprints ne nous intéressent pas beaucoup. Avec iceScrum nous allons quand même créer un sprint, mais de longue durée, par exemple 90 jours.

On pourra toujours modifier la date de fin de sprint plus tard, par exemple pour la prolonger.

Le sprint est créé à la création du projet ou plus tard, à partir du plan de release.

On peut aussi le faire, lors de l'assistant de création ou en accédant aux pratiques du projet, définir une durée longue, comme ci-dessous :

A screenshot of a 'Practices' configuration window. The window has a title bar 'Practices' and a dropdown menu set to '...for planning'. Below the dropdown are four rows of settings:

- 'Suite used for estimation' with a dropdown menu showing 'Fibonacci'.
- 'Sprint duration (days)' with a text input field containing '90'.
- 'Disable Estimation (effort = 1)' with radio buttons for 'Yes' and 'No', where 'No' is selected.
- 'Hide weekends in charts' with radio buttons for 'Yes' and 'No', where 'No' is selected.

Below these settings are three expandable sections: '...for sprint management', '...for urgent tasks management', and '...for meeting hours management'. At the bottom right of the window are 'Update' and 'Cancel' buttons.

## Création d'éléments

Dans Scrum, et dans iceScrum, on distingue la story de la tâche. La tâche est un travail qui contribue à la réalisation d'une story. Cette distinction est très utile dans les projets de développement, mais avoir ces 2 notions n'est pas nécessaire pour gérer des petits travaux orientés flux.

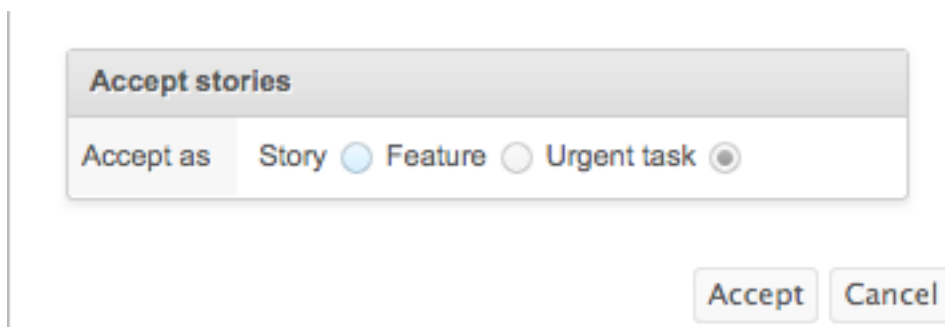
Dans notre approche ScrumBan, nous allons utiliser intensivement la notion de tâche, et avec iceScrum en particulier, la tâche urgente. Nos travaux seront des tâches urgentes pour iceScrum.

Il est possible de créer des tâches urgentes directement dans la vue Plan de sprint.

## Utilisation du bac à sable

Une autre possibilité, plus adaptée à la collecte de travaux divers venant de sources différentes, consiste à créer une story dans le bac à sable. Bien que nous ne souhaitions pas utiliser la notion de story, le bac à sable offre une ouverture vers le monde extérieur et des possibilités de discussion bien supérieures à celles du plan de sprint et l'élément qu'on y met est appelé, pour des raisons historiques, une story.

Une fois les «stories» entrées dans le bac à sable et après un éventuel dialogue avec les créateurs, le Product Owner peut les accepter comme ... tâches urgentes (cette possibilité n'est proposée que si un sprint est activé).



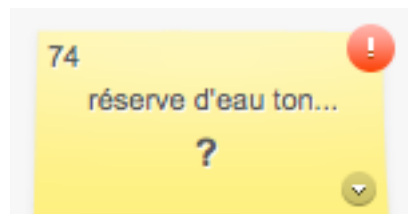
## Utilisation du plan de sprint

Le plan de sprint est réduit à une seule ligne, celle des tâches urgentes, faisant apparaître la limite, définie précédemment, pour l'état en cours.

Les tâches créées sont visibles dans la colonne A faire. Une fois le sprint activé, on peut les déplacer dans la colonne En cours.

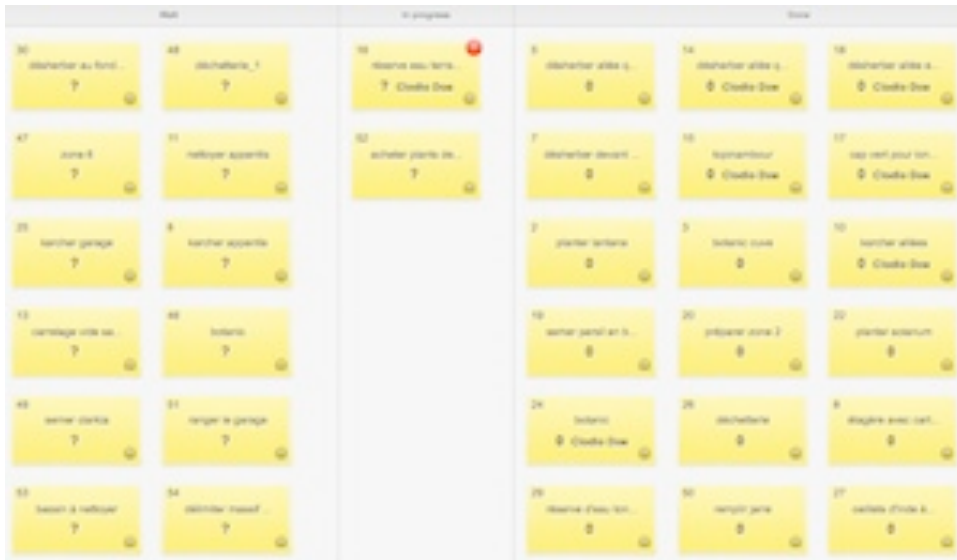
Le principe fondamental du kanban est la limitation du travail en cours. La limite définie dans iceScrum empêche de la violer : l'ajout d'une tâche dans la colonne à faire, alors que la limite est atteinte, est refusé.

Si une tâche ne peut pas se finir normalement, on peut le faire apparaître visuellement en la déclarant comme bloquée. De la même façon, avec la commande Bloquer, on peut mettre en évidence les tâches à faire qui ne peuvent pas démarrer.

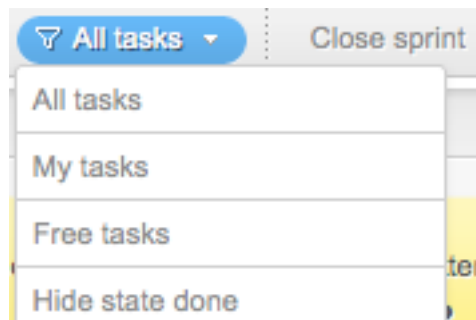


Avec cette approche, une estimation en heures des tâches est généralement inutile. Cependant, avec iceScrum il reste possible d'indiquer le reste à faire pour chaque tâche.

Comme ce tableau a une longue durée de vie (par rapport à un sprint classique avec Scrum), les tâches finies vont s'amonceler dans la colonne de droite.



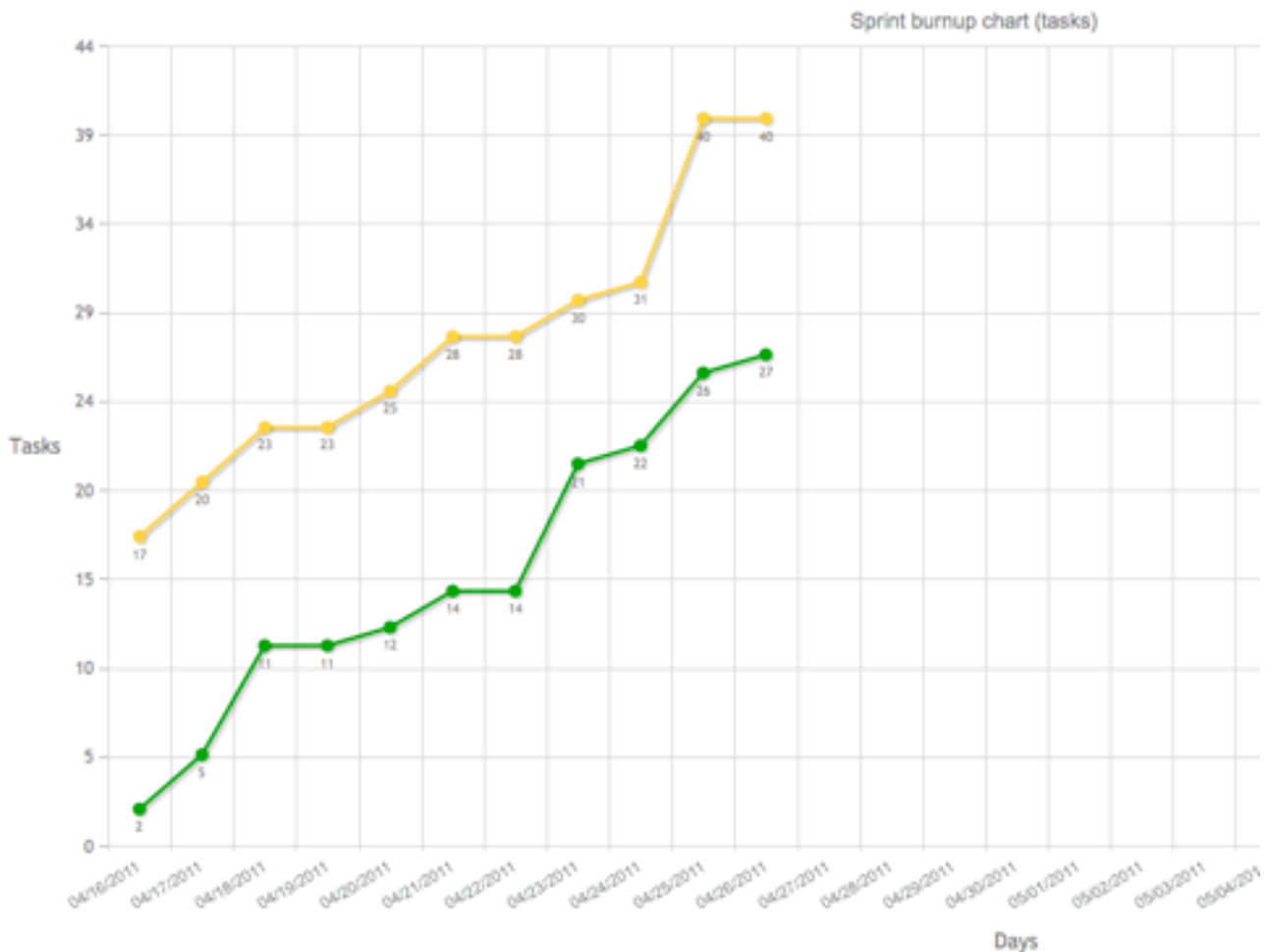
Pour y voir plus clair dans les 2 autres colonnes, il est possible de masquer les tâches finies, avec la commande suivante :



## Indicateurs

Les indicateurs portant sur les stories ou sur les features ne seront d'aucun intérêt dans cette approche. Même si on estime les tâches en heures, le burndown de sprint en heures ne sera pas significatif, puisqu'on est dans un contexte où il n'aura pas nécessairement une tendance à descendre.

L'indicateur le plus intéressant sera le burnup en tâches, qui montre 2 courbes : une pour le total de tâches et l'autre pour celle qui sont finies, actualisées tous les jours.



## Un peu de Scrum

Avec cette approche ScrumBan, on conserve la possibilité d'utiliser quelques pratiques Scrum dans iceScrum :

- le rôle de Product Owner,
- la définition de fini pour indiquer la stratégie d'utilisation du kanban,
- la release, comme jalon. Un exemple est d'avoir une release de 3 mois avec un seul sprint, nécessaire pour avoir un plan de sprint kanbanisé.

Il reste bien entendu possible d'ajouter des stories dans le sprint, en conjonction avec cette approche centrée sur les tâches.

## Evolutions dans iceScrum pour un meilleur ScrumBan

On a vu une utilisation très simple d'iceScrum qui peut convenir quand les travaux à suivre rentrent bien dans ce workflow basique.

Quelques petites évolutions pourraient améliorer cet usage :

- avoir plusieurs lignes dans le tableau (une possibilité déjà offerte dans iceScrum est d'utiliser la ligne pour les tâches récurrentes, mais elle ne gère pas de limite)
- ajouter le diagramme de flux cumulé dans les indicateurs du sprint, qui serait actualisé tous les jours (le burnup de tâches montre déjà 2 états : fini ou pas fini, qui est le total de à faire et en cours)

Evidemment une amélioration intéressante dans l'esprit Kanban serait de pouvoir modifier le workflow. Dans l'approche proposée ici, on a un workflow imposé, à 4 états :

- en attente dans le bac à sable
- à faire
- en cours
- fini

Pour déjà mieux visualiser ce workflow, qui est sur 2 vues, une évolution dans iceScrum va permettre de voir le bac à sable en vue réduite sur la gauche (widget) en même temps que le plan de sprint.

Sur ces 4 états, l'outil gère le flux descendant, avec la limite de TAF uniquement sur l'état en cours.

Une mesure essentielle en kanban est le temps de cycle. iceScrum a déjà, en plus du burnup, les outils permettant de le calculer. Pour une tâche, on conserve la date de changement d'état :

.....  
**Creation date** : 04/21/2011 14:36:33  
.....

**In progress date** : 04/25/2011 08:51:57  
.....

**Done date** : 04/25/2011 16:43:12  
.....

Pour les tâches urgentes créées en dehors du plan de sprint, il suffirait d'y ajouter la date de mise dans le bac à sable.

L'outil autorise actuellement du flux remontant : une tâche en cours peut repasser dans la colonne à faire (mais une tâche finie ne peut pas remonter le flux). Une option pourrait être ajoutée dans la configuration des pratiques, pour l'autoriser ou pas.