

# **Quelques outils pour organiser un projet agile**



# Table des matières

<b>I - Expliciter la méthode suivie</b>	<b>3</b>
<b>II - Espaces de discussion</b>	<b>4</b>
<b>III - Product backlog &amp; user stories</b>	<b>5</b>
<b>IV - Kanban</b>	<b>6</b>
<b>V - Journal de bord</b>	<b>7</b>
<b>VI - Fiche de temps</b>	<b>8</b>
<b>VII - Documentation</b>	<b>9</b>
<b>VIII - Exercice : Défi</b>	<b>10</b>
<b>Contenus annexes</b>	<b>12</b>

# Expliciter la méthode suivie

---



Une gestion agile ce n'est pas « on fait comme on veut », c'est « voici les outils qui nous permettront de collaborer dynamiquement (de piloter) »



- Si vous suivez une méthode standard, indiquez-le et précisez selon quelles modalités.
- Sinon, explicitez votre mode de fonctionnement.

# Espaces de discussion

---



- Indiquez quelles modalités de discussions sont utilisées (renvoyez vers les éventuels tutoriels et guides de bonnes pratiques associés).
- Si plusieurs modalités sont utilisées (*chat*, mail, téléphone...), précisez quelles modalités sont favorisées pour quels usages.

# Product backlog & user stories

---



**Rappel**

---

*Exemple d'outils agiles (cf. p.12)*



**Attention**

---

Pensez aux livrables, chaque tâche doit produire des choses à voir et chaque *time box* doit consister à parler de livrables (on ne raconte pas, on donne à voir, on parle des livrables).

# Kanban



Le *product backlog* peut être géré avec un outil de kanban numérique.

## Exemple d'organisation typique pour un kanban

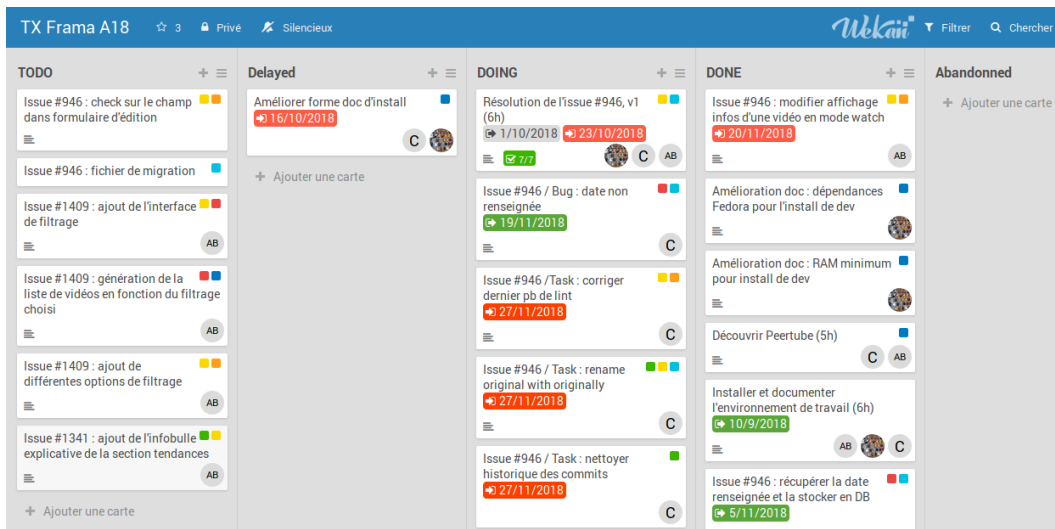


Il se présente sous la forme d'un tableau à 5 colonnes proposant un instantané des tâches composant le projet :

- TODO : toutes les tâches encore à faire
- Delayed : les tâches prévues à un moment, et reportées entre temps
- DOING : les tâches en cours de traitement sur l'itération
- DONE : les tâches terminées
- Abandoned : les tâches abandonnées



Wekan est un kanban hébergé par Picasoft : [kanban.picasoft.net](https://kanban.picasoft.net)<sup>1</sup>



<sup>1</sup> <https://kanban.picasoft.net/>

# Journal de bord



Le journal de bord est un enregistrement séquentiel de ce qui est réalisé, avec des notes permettant de garder la mémoire des réflexions associées.

Il contient des références vers les livrables.

Il permet également de gérer les réunions (ODJ et CR).



- Vérifier l'existence des services définis dans le fichier `json` (existence du dossier à sauvegarder)
- Créer/Éditer une entrée sur le Wiki permettant de documenter le travail effectué

Auto-évaluation (Alexandre-Guillaume) : 3/3

Auto-évaluation (Samuel) : 3/3

Avis des suiveurs : 3/3 pour les deux.

## Semaine 5

### Améliorations

- Réécriture du `Dockerfile` de `backup-rotation` pour prendre en compte les "best practices" Docker.
- Suppression du tag `latest` pour l'image de base (instruction `FROM`)
- Ajout de commentaire dans le script Python `start_rotation.py` (rappel : script qui permet l'ajout des tâches `cron` après lecture du fichier de configuration `backup_data.json`)
  - Les commentaires en Français c'est pas trop grave, par contre il faudrait éviter les accents. On est jamais à l'abri (comme ici) de tomber sur une machine qui a un e
- Modification de l'image `pica-backup` pour pouvoir traiter le fichier `json` utilisé par `pica-backup-rotation` afin de centraliser l'information

### Tests

- Test du conteneur `backup-rotation` sur de faux fichiers de sauvegarde générés par `fake_backups.py` (les 50 000 fichiers générés ont été réduits à 46 -> comportement)
- Test de la nouvelle image `pica-backupv2` (voir ci-dessous)

### Nouveaux livrables

- Écriture d'un script Python pour la génération de fichiers vides comportant les mêmes noms que les fichiers de sauvegarde d'un service. On appelle le script de la man sauvegarde faite toutes les heures en remontant dans le temps jusqu'à atteindre le nombre de sauvegardes indiqué en paramètre. Le format du nom du fichier est : `YY`
- Écriture du fichier `README`. Fichier : `/root/samuel/pica-backup-rotation/README.txt`
- Écriture du script `/root/samuel/pica-backup/backup_env_var.py` permettant de transformer les paramètres du fichier `backup_data.json` en variable d'environnement `t`
- Création de l'image `pica-backupv2` qui va lire les paramètres de configuration des sauvegardes dans le fichier de configuration `backup_data.json` (même fichier que `p`)

### Modifications du wiki

<https://wiki.picasoft.net/doku.php?id=bs:infra-p18>

- Écriture d'une documentation sur la configuration et fonctionnement du conteneur `backup-rotation`
- Documentation sur le test du conteneur de rotation

Auto-évaluation (Alexandre-Guillaume) : 3/3 validé !

Auto-évaluation (Samuel) : 3/3 validé !

## Semaine 6

### Améliorations

- Prise en compte des commentaires d'Antoine et Yuena sur la MR

# Fiche de temps



## Définition

La fiche de temps permet de suivre régulièrement le travail réalisé dans le cadre du projet. Elle est mise à jour au fur et à mesure.

La fiche de temps est avant tout un outil d'auto-contrôle permettant notamment de détecter des tâches qui dérivent.



## Exemple

	A	B	C	D	E
1		<i>Total</i>	83	101,75	
2	<b>Date</b>	<b>Tache</b>	<b>Temps de travail (h) AR</b>	<b>Temps de travail (h) MD</b>	<b>Commentaires</b>
3	02/03	Réalisation minutieuse du module 1 en apprenant		1	Problème avec lien, paramétrage de VM et installation par réseau
4	03/03	Recherche de doc et de solutions pour le module 1		1,5	Problème dû à la non-activation du VT-x notamment
5	04/03	Premières corrections du module 1		2	Prise en main fastidieuse d'Opale et premières corrections notamment via captures d'écrans
6	04/03	Prise en main Module 2 en apprenant		1	/
7	05/03	Capture d'écrans et modifications textes module 1 (Debian)		3	Voir si captures supplémentaires nécessaires
8	06/03	Prise en main	2		Page wiki sr-p18 inexistante, à initialiser ? Mismatch modules hébergés/publiés?
9	07/03	Initialisation du wiki	0,5		
10	11/03	Finition module 1		2	
11	11/03	Correction module 2		2	fini
12	11/03	Relecture et finition module 1/2	3		Pas d'ajout
13	11/03	Prise en main et repérage de corrections à effectuer		1,5	
14	12/03	Correction module 3		0,5	Revérifier si le multisite fonctionne bien
15	12/03	Prise en main module 4		0,5	Niveau technique semble ok mais rajouter des informations sur ce qu'est le php
16	12/03	Prise en main et amélioration module 5	3		Ajout de glossaire, correction typo, liaison entre sections
17	12/03	Prise en main module 4	1		



# Documentation

---



Il est fondamental que toute la documentation produite soit correctement gérée, c'est à dire que l'on sache facilement :

- quels documents existent
- où ils se trouvent
- qui peut les modifier et comment
- quel est leur statut (en projet, en cours, finalisé...)
- quel est leur spectre de diffusion (confidentiel, diffusion ouverte sous copyright, licence libre...)

La gestion des documents produits pendant le projet peut se faire selon plusieurs moyens :

- espace partagé (exemple : NextCloud)
- système de gestion de version pour des fichiers textuels (exemple : Git + markdown)
- wiki (exemple : MediaWiki)
- CMS web (exemple : WordPress)
- GED, chaîne éditoriale collaborative (exemple : Scenari)

# Exercice : Défi

---



Vous êtes membre d'une association (celle de votre choix).

L'outil Mobilizon ([joinmobilizon.org](https://joinmobilizon.org)<sup>1</sup>, [framapiaf.org/@mobilizon](https://framapiaf.org/@mobilizon)) de Framasoft va sortir d'ici quelques jours... Il est déjà disponible en test.

Vous êtes en charge de la gestion d'un projet en mode agile pour préparer une campagne d'information pour un événement de votre association qui utilisera (par exemple une journée thématique organisée à l'UTC en présentiel et ligne).

## Question 1

Produisez une page web qui décrit votre méthode : *time boxes*, organisation, outils utilisés...

### Indice :

[aswemay.fr/co/methodo-tx.html](http://aswemay.fr/co/methodo-tx.html)<sup>2</sup>

## Question 2

Produisez une page web *Modalités de communication* qui explicite, par exemple, que l'on utilisera Mattermost sur le serveur de Picasoft et le mails. Renvoyez aux documents afférents, explicitez quand on utilise quelle modalité.

## Question 3

Produisez un pad qui contient au moins 3 *users stories*

## Question 4

Produisez un kanban qui représente un *product backlog* qui contient au moins 6 tâches.

Vous utilisez [kanban.picasoft.net](https://kanban.picasoft.net)<sup>3</sup> et vous ajouterez votre enseignant au projet, avec les droits de votre choix (au moins la lecture donc) ?

## Question 5

Produisez un pad qui contient votre journal de bord (avec au moins une entrée pour chacune des 5 tâches réalisées dans le cadre de cet exercice)

## Question 6

Produisez un pad ou une page web qui contient votre fiche de temps à jour en fin d'exercice.

---

<sup>1</sup> <https://joinmobilizon.org/fr>

<sup>2</sup> <http://aswemay.fr/co/methodo-tx.html>

<sup>3</sup> <https://kanban.picasoft.net>

## **Question 7**

Produisez une page web qui décrit votre politique de gestion documentaire (choisissez une ou plusieurs modalités de documentation, vous n'avez pas à les mettre en place dans le cadre de cet exercice, décrivez simplement comment on doit faire en supposant les outils disponibles).

## **Question 8**

Produisez une page web portail qui contient tous les liens vers les outils ci-avant.



# Contenus annexes

## 1. Exemple d'outils agiles

### Timeboxing



Principe du *timeboxing* : Les itérations ont une durée fixe, sont à court terme (< 1 mois), on évalue chaque itération à la fin et on réajuste le plan de travail en fonction des résultats.

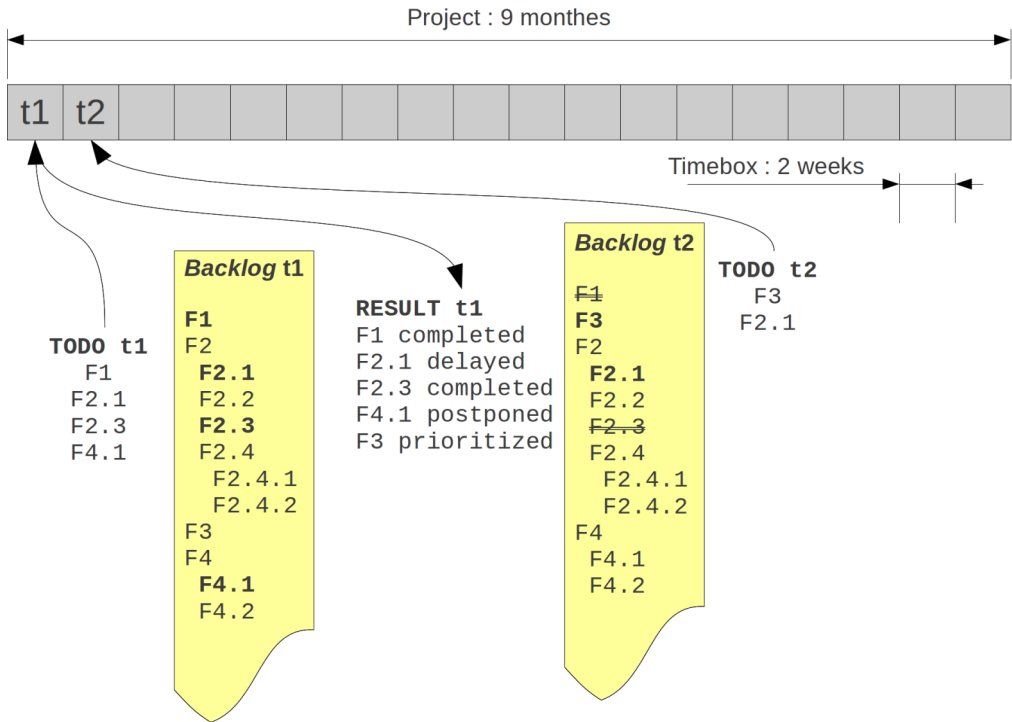
### Product backlog



On établit en début de projet la liste des besoins exprimés, par exemple sous forme fonctionnelle. Ces fonctions sont hiérarchisées par facteur d'importance en terme d'usage et par délais, difficulté et/ou coût de réalisation. C'est le *product backlog*.

Avant chaque *timebox* le *product backlog* sert d'outil d'arbitrage, il permet de commencer par le plus important, le moins coûteux, le plus urgent, le plus simple, ...

Après chaque itération le *product backlog* est mis à jour : fonctions réalisées, évolution de l'importance relative d'un besoin, évolution du coût de réalisation étant donné le chemin parcouru ou l'apparition d'une nouvelle technologie, ...



Timeboxing & product backlog

## User stories



Il s'agit de raconter en petites histoires d'une dizaine de lignes environ, les principaux scénarios d'utilisation illustrant l'usage du système visé.

Ces histoires permettent d'avoir un fil conducteur mental pour **visualiser** les fonctions pertinentes. Elles doivent être rédigées (en français !) et ne pas se réduire à un simple schéma ou des listes à puces. C'est le fait d'écrire qui permet de penser au scénario dans son déroulement temporel, de penser la cohérence des actions et processus dans leur agencement et succession.

Ces scénarios se construisent souvent autour des rôles des utilisateurs : administrateur, usager standard, usager expert, etc.

Si un schéma appuie l'histoire, il ne doit pas la remplacer. Un scénario n'est pas un *use case* (au sens UML), justement parce que c'est rédigé et non schématique.

## Transcription d'une vidéo



« Valérie est en charge de la transcription et de la synchronisation de la collection des 8 entretiens de metteurs en scène. Cette opération préalable à la mise en ligne d'un entretien est longue et fastidieuse. Elle compte ainsi profiter des nouveaux outils pour accélérer cette étape de traitement. Elle est averti de la mise à disposition de l'entretien de Roger Planchon pour l'étape de transcription/synchronisation. Elle consulte alors le document et peut commencer à taper au kilomètre le texte brut. Valérie peut ajouter quelques éléments de typage (important) dans le corps de son texte. A chaque frappe, le player vidéo se met en pause et reprend, une fois la frappe terminée, quelques secondes auparavant. Cette méthode de saisie est très appréciée par Valérie qui peut ainsi écouter un fragment puis taper le texte sans avoir à mettre constamment en play/pause le curseur vidéo, ni à revenir avec la barre de navigation.

(par Ludovic Gaillard, INA, projet C2M, <http://www.utc.fr/ics/c2m>)

