

# Plan de Développement

<b>Version</b>	0.1
<b>Date</b>	16 janvier 2012
<b>Rédigé par</b>	Claire SMETS
<b>Relu par</b>	Emmanuel MOCQUET

## MISES À JOUR

Version	Date	Modifications réalisées
0.1	11/12/2011	Création
0.2	07/01/2012	Modifications et ajouts
1.0	16/01/2012	Première version complète

## Table des matières

1	Contexte du projet	4
2	Documents applicables et de référence	5
3	Terminologie et sigles utilisés	5
4	Méthodologie de développement	5
5	Organisation et responsabilités	7
6	Organigramme des tâches	8
7	Évaluation du projet et devis technique	9
8	Dimensionnement des moyens	9
9	Planning général	11
10	Procédés de gestion	12
10.1	Gestion de la documentation . . . . .	12
10.2	Gestion des configurations . . . . .	12
11	Revue et points clefs	12
12	Procédures de suivi d'avancement	12

## 1 Contexte du projet

De nos jours les capacités des ordinateurs croissent de manière exponentielle, tant au niveau des performances de calcul qu'au niveau des capacités de stockage. Pour ce faire, de nouveaux algorithmes et technologies sont constamment inventés et mis à disposition du public.

La mémoire flash contenue dans les SSD (Solid-State Drive) est l'une des technologies relativement récentes comportant de nombreux avantages dont les suivants :

- le temps de lecture et d'écriture des données est bien plus court : à la différence d'un disque dur "classique" il n'y a pas besoin de rotation du plateau afin d'aller chercher les informations nécessaires. Le temps d'accès à une zone mémoire est toujours le même.
- une économie d'énergie importante : l'énergie habituellement dépensée à faire tourner les disques n'est plus nécessaire.
- l'usure mécanique est inexistante : il n'y a pas de disque à faire tourner ni de bras qui frotte.

Cependant il y a aussi quelques inconvénients :

- le prix d'1Go de mémoire flash reste assez élevé et le prix peut monter très haut pour des capacités aujourd'hui considérées comme raisonnables pour un disque dur à mécanique.
- le nombre de cycles d'écritures est limité (quelques centaines de milliers) alors qu'il est illimité pour les autres.

Tout cela conduit à l'idée d'un nouveau système qui serait un compromis entre la rapidité et l'économie d'énergie de la mémoire flash d'une part, et la capacité de stockage à moindre coût ainsi qu'une plus longue durée de vie d'autre part.

L'objectif de projet est donc la mise en place d'un tel système hybride. À partir d'un périphérique de stockage de masse "classique" et un autre avec de la mémoire flash, créer un nouveau périphérique bloc virtuel qui formera une couche entre les deux périphériques et le noyau. Le nouveau Périphérique Bloc Virtuel s'occupera de la gestion de l'ensemble de l'espace de stockage nouvellement formé.

Ce projet est réalisé dans le cadre de l'enseignement de première année de master Sécurité des Systèmes Informatiques. Il sera réalisé par une groupe de six étudiants. La première partie du projet consiste simplement en la rédaction des documents qui nous ont permis de mieux définir le sujet, les objectifs, les risques, et l'organisation du projet.

La seconde partie est le développement du code du projet. Celui-ci sera jalonné par plusieurs soutenance qui nous permettront de nous assurer de toujours aller dans la bonne direction.

Notre client, M.Nicart, est maître de conférences en informatique scientifique. Cela facilitera les échanges entre lui et notre équipe afin de toujours s'assurer d'être en accord.

## 2 Documents applicables et de référence

**Appel d'offre :** Projet1-2-3.pdf

**Spécification Technique du besoin :** PBV\_STB\_1.0.pdf

**Architecture Logicielle :** PBV\_DAL\_1.0.pdf

**Cahier des Recettes :** PBV\_CdR\_1.0.pdf

**Analyse des risques :** PBV\_AR\_1.0.pdf

**Plan de développement :** PBV\_PdD\_1.0.pdf

**Planning général :** PBV\_PG\_1.0.pdf

## 3 Terminologie et sigles utilisés

**Module noyau :** Programme qui peut être chargé dynamiquement dans le noyau.

**SSD :** Solid State Disk, périphérique de stockage qui utilise de la mémoire flash.

Les performances sont élevées et les consommations d'énergie basses mais le coût par Go est élevé.

**HDD :** Hard Drive Disk, périphérique de stockage de masse.

Il consomme plus et ses performances sont moindres que celles d'un SSD mais le prix d'acquisition est beaucoup moins élevé.

**PBV :** Périphérique Bloc Virtuel. Il s'agira ici d'un fichier représentant deux disques (SSD et HDD).

Il est cependant possible qu'il ne soit composé que du HDD.

**SGF :** Un Système de Gestion de Fichiers "est une façon de stocker les informations et de les organiser dans des fichiers". (Wikipedia)

**Pilote :** Programme gérant les opérations entre un périphérique et le reste du système.

**Mémoire cache :** Mémoire volatile d'accès rapide. Les données ne sont pas liées à un processus en cours d'exécution.

**Swap :** Zone mémoire réservée sur l'espace de stockage de masse et/ou sur le PBV servant de mémoire d'appoint à la mémoire principale.

## 4 Méthodologie de développement

Dans ce projet nous utiliserons une méthode agile et plus précisément la méthode XP (Extreme Programming) : elle pousse à l'extrême des principes simples et impose une discipline de développement afin que celui-ci se passe dans les meilleures conditions.

La principale caractéristique de cette méthode est l'implication constante du client afin d'être sûrs de faire évoluer le projet dans le sens de ses attentes.

Elle repose sur cinq valeurs fondamentales :

**La communication** Elle évite une incompréhension à tous les niveaux (client/fournisseurs, testeurs/programmeurs, ...) et permet par conséquent de ne pas perdre de temps ni d'énergie à aller dans une mauvaise direction. Il est donc conseillé de présenter très régulièrement les avancées du projet (même petites) au client. De plus au sein du groupe il est important d'avoir de nombreux échanges et le travail en sous équipes est fortement privilégié.

**La simplicité** Ne pas produire une application que l'on aura du mal à faire évoluer : développer un code simple qui sera en perpétuelle évolution et qui pourra être optimisé et/ou implanter de nouvelles options.

**Le feed back** Toujours savoir où en est l'avancement du projet, si les tests sont valides et dans le cas contraire les problèmes rencontrés. Ils permettent de connaître l'avancement dans le projet.

Avant chaque fonctionnalité implantée des tests seront écrits. Et à chaque intégration de code (ou toute modification) TOUS les tests seront systématiquement relancés. Cela implique une intégration en continu. Cela permet de ne pas se heurter à une multitude d'incompatibilités au dernier moment. Et cela permettra aussi de pouvoir très régulièrement présenter une version fonctionnelle du projet au client.

**Le courage** Être capable de jeter du code et de repartir dans une autre direction. Le feed back et la communication y aident.

**Le respect** L'équipe est responsable de tout le code et peut y intervenir à tout moment et les tests permettent d'indiquer si quelque chose a été cassé. Par conséquent il est indispensable de définir des règles de codage afin que chacun s'y retrouve.

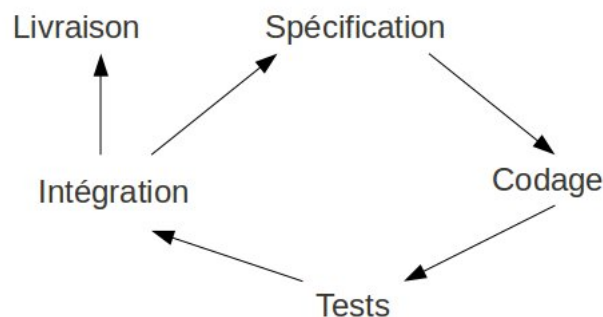
Nous nous efforcerons de coller au mieux à ce modèle.

En résumé, différentes versions du projet seront régulièrement présentées au client. Chaque version validera tous les tests développés : chaque fonctionnalité implantée sera tout de suite intégrée au projet et tous les tests seront relancés. Tant que tous les tests ne seront pas valides la version ne pourra être présentée au client. Une réunion sera programmée avec le client à chaque présentation d'une nouvelle version.

Le développement s'effectuera toujours en binôme afin de toujours être dans une optique de communication :

1. Zakaria ADDI et Zineb ISSAAD
2. Baptiste DOLBEAU et Emmanuel MOCQUET
3. Pierre LUCE VEYRAC et Claire SMETS

Et s'effectuera selon le schéma suivant :



**La spécification** permettra de déterminer quelles fonctions seront à écrire, leurs préconditions et postconditions.

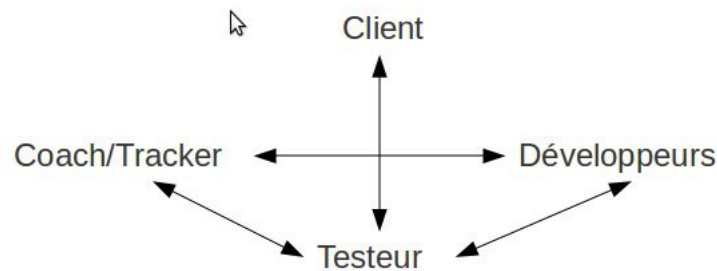
**Le codage** implémentera les spécifications précédemment définies

**Les tests** permettront de s'assurer le code produit est correct

**L'intégration** entraînera l'incrément de la version et un produit qui sans bug qui comporte de nouvelles fonctionnalités.

**La livraison** permettra d'avoir les opinions du client quant à la nouvelle version.

## 5 Organisation et responsabilités



Lors de ce projet chaque membre de l'équipe a un rôle particulier.

**Les programmeurs** sont responsables de la production du code. Ils aident aussi à la réestimation de la charge de travail en fonction de l'avancement du projet. Ils seront en dialogue permanent avec le client. Ces rôles sont tenus par Baptiste DOLBEAU, Pierre LUCE VEYRAC et Emmanuel MOCQUET.

**Le client** communique les informations utiles aux développeurs. Il assure l'intermédiaire entre le demandeur et le groupe. Ce rôle est tenu par Zakaria ADDI.

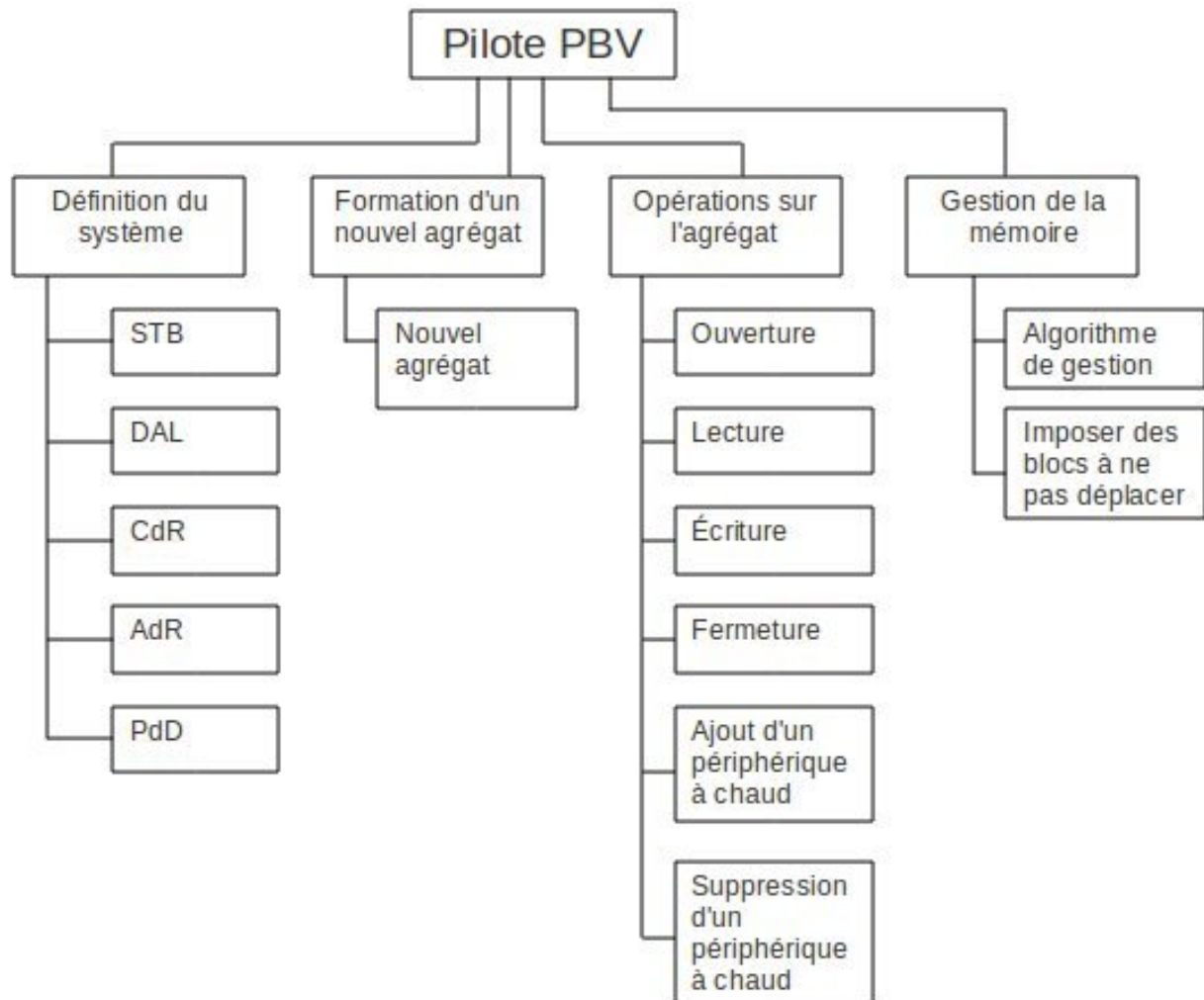
**Le testeur** conçoit et réalise les tests des fonctionnalités qui seront implantées. Ces tests devront couvrir le plus large panel de situations possible. Il témoignera aussi de l'avancement du projet. Ce rôle sera tenu par Zineb ISSAD.

**Le tracker** s'assure de l'avancement du projet et du respect des délais. Il essaie aussi de détecter les difficultés le plus tôt possible. Ce rôle sera tenu par Claire SMETS.

**Le manager** ne fait pas partie de l'équipe. Il nous fournit le matériel dont nous avons besoin (SSD) et supervise l'ensemble du projet.

**Le coach** rend compte au manager de l'évolution du projet. Il s'assurera du bon fonctionnement de l'équipe (communication, chacun assure sa fonction). Ce rôle sera tenu par Claire SMETS.

## 6 Organigramme des tâches





## 7 Évaluation du projet et devis technique

**Version 0.1** Elle consistera simplement à former un nouvel agrégat à partir des deux périphériques branchés (un SSD et un HDD exactement). Aucune opération ne pourra être effectuée dessus. Les deux périphériques de stockage de masse seront simplement vus comme un seul et même espace de stockage.

**Version 0.2** Elle ne sera pas non plus fonctionnelle. Elle permettra d'ouvrir et de fermer le nouvel agrégat que l'on pourra former à partir des deux périphériques. Aucune fonction ne sera encore possible.

**Version 0.3** Elle permet de lire et d'écrire des blocs sur l'agrégat, comme sur un périphérique non hybride.

**Version 0.4** Elle permet de gérer l'allocation des blocs de l'agrégat : les blocs souvent lus ou modifiés seront prioritairement placés sur le SSD tandis que ceux moins ouverts et modifiés seront peut-être "poussés" vers le HDD.

**Version 0.5** Les fonctionnalités implantées à partir de cette version sont optionnelles et dépendent du bon avancement du projet. Cette version permettra à l'utilisateur de pouvoir choisir des fichiers à ne pas déplacer du SSD ou du HDD.

**Version 0.6** Elle permettra de pouvoir ajouter ou supprimer des périphériques à chaud : jusqu'à maintenant, l'agrégat ne comportait qu'un seul SSD et un seul HDD, tous deux branchés avant l'allumage de la machine, et débranchés après. Cette version permettra de pouvoir changer les périphériques pendant que la machine est allumée, et que l'agrégat soit différemment composé (un ou plus HDD avec zéro ou un SSD). À noter que le projet perd son intérêt si les deux types de périphériques de stockage ne sont pas présents. Mais ce peut être une option.

## 8 Dimensionnement des moyens

Pour ce projet, nous avons besoin de matériel : un SSD et un HDD afin d'effectuer les tests. Le langage de programmation étant le C et l'assembleur, aucun logiciel spécifique n'est requis : le développement pourra se faire avec des outils tels que Geany, Vi, Gedit, Emacs, ... à la préférence des développeurs.

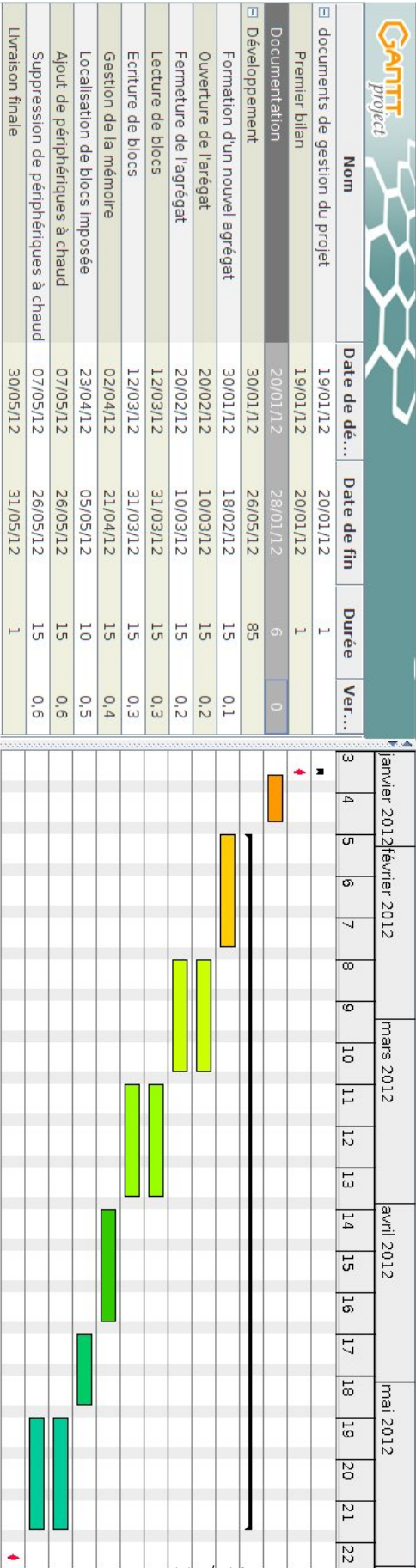
Cependant le pilote ne sera valide que sur les OS à noyau Linux. Par conséquent chacun devra avoir une machine avec un OS répondant à ce critère.

Le code produit ainsi que les différentes versions ainsi que les documents produits seront stockés sur Github, à l'adresse [https://github.com/Maguy/gestion\\_projet/tree/](https://github.com/Maguy/gestion_projet/tree/).

Du fait d'avoir besoin d'un SSD et d'un HDD pour les tests, ceux-ci ne pourront s'effectuer que sur une machine de tests, à l'université.



9 Planning général



## 10 Procédés de gestion

### 10.1 Gestion de la documentation

Les documents de référence pour la gestion de ce projet seront disponibles au format pdf sur l'espace de stockage créé sur GitHub ([https://github.com/Maguy/gestion\\_projet/tree/](https://github.com/Maguy/gestion_projet/tree/)). Chaque membre responsable d'une fiche la fera évoluer au fil du projet, et incrémentera le numéro de version :

- la Spécification Technique du Besoin : Zakaria ADDI
- le Document d'Architecture Logicielle : Emmanuel MOCQUET
- l'Analyse des Risques : Baptiste DOLBEAU et Pierre LUCE VEYRAC
- le Cahier des Recettes : Zineb ISSAD
- le Plan de Développement : Claire SMETS

L'ensemble du groupe devra se tenir au courant de leur évolution.

De plus après chaque réunion avec le client, un compte rendu sera rédigé et validé par l'équipe avant d'être soumis à validation par le client, puis stocké sur l'espace de stockage partagé.

### 10.2 Gestion des configurations

Pour chaque nouvelle fonctionnalité une branche sera créée sur l'espace de stockage afin de stocker le code produit. Chacun pourra modifier le code de tout le reste de l'équipe, mais devra le signaler dans l'entête du fichier modifié et le numéro de version sera incrémenté. Chaque fonction sera commentée afin de faciliter la relecture et le débogage du code.

## 11 Revues et points clefs

Première revue du projet : le jeudi 19 janvier.

Les documents de référence auront été rendus au plus tard le lundi 16 janvier, et des experts en conduite de projets les annoteront. Suite à ça l'équipe soutiendra devant un jury composé des experts et du client afin de juger de l'avancement du projet.

## 12 Procédures de suivi d'avancement

Toutes les semaines une réunion aura lieu entre les membres de l'équipe afin de juger de l'état d'avancement du projet et d'échanger sur les difficultés rencontrées et les solutions apportées. Chaque membre de l'équipe parlera de l'avancement de la tâche qui lui a été assignée, et un bilan de l'avancement du projet en fonction du planning sera fait.

De même à chaque livraison une réunion sera aussi organisée avec le client afin de s'assurer de toujours être en accord sur l'évolution du projet. Par conséquent les livraisons seront assez régulières.

Des jalons seront posés afin de juger de la vitesse à laquelle avance le projet, et de ne pas se laisser surprendre par une date buttoir. En cas de retard, une décision sera prise par le coach en accord avec les autres membres de l'équipe, et le planning sera réajusté.

Le planning sera dans tous les cas continuellement mis à jour.