

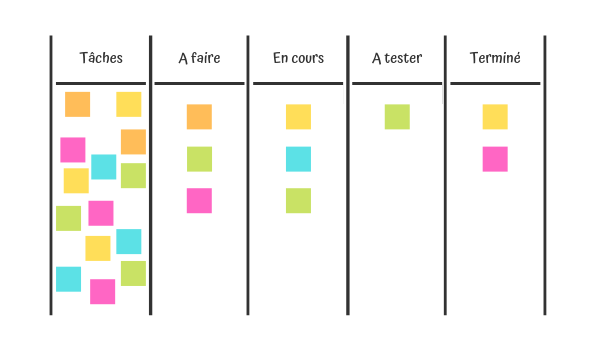
|  |
| --- |
| SIT213 BE - 1  IMT Atlantique |
| Chaire M@D - Maintien à domicile | IMT Atlantique |
| 14 septembre  Créé par :  LE DUC Elouan MAQUIN Philippe LE GRUIEC Clément LE JEUNE Matthieu FRAIGNAC Guillaume |

# Organisation du travail d’équipe

Dans cette présentation nous allons exposer les choix que nous avons sur notre organisation de groupe.

Pour respecter les délais nous avons choisis d’utiliser un tableau de kanban. Ce tableau utilise des « posthites » pour visualiser de manière très simple l’évolution du travail au cours du temps.

Voici un exemple d’un tableau de Kanban.



Pour garantir la qualité de nos livrables, nous allons nous focaliser sur les points suivants :

La forme : nos livrables devront être cohérents en mise en page (utilisation de la même police, même page de garde, etc.).

Le contenu doit être assez précis pour un lecteur expert du domaine, mais accessible pour un lecteur qui ne l’est pas.

Les tests sont une partie très importante ; en effet, ils nous permettent de vérifier le fonctionnement de notre code. Ainsi les tests doivent être assez exhaustifs pour couvrir toutes les fonctionnalités.

Pour assurer la qualité de notre documentation nous avons mis en place des procédures de relecture. Avant de rendre un livrable, nous effectuons plusieurs relectures par différentes personnes. Ainsi nous pouvons nous assurer d’une certaine qualité de documentation. Nous apportons également une importance particulière au respect des règles de bonne pratique du génie logiciel.

Pour synchroniser le travail, nous allons utiliser un outil de versioning ( Git ). Nous avons choisi GIT pour plusieurs raisons : ce logiciel nous permet de créer plusieurs branches ; nous avons décidé de créer une branche pour chaque membre de l’équipe dans laquelle il aura un espace dédié pour travailler. Grâce à la fonction reverse de GIT, nous pouvons rattraper des erreurs commises.

Une fois notre version validée par l’équipe, nous pouvons merger notre code dans la branche master qui nous servira de point de départ pour la prochaine étape du développement.

Afin que GIT ne devienne pas un endroit de non-lieux, nous avons imposé certaines règles :

* Des noms explicites pour chaque commit
* Faire des commit réguliers
* Le code doit au minimum correctement compiler
* Utiliser des tags ( add et remove ) pour informer des modifications dans les commit

Nous utiliserons des logiciels de test ( JUnit et Emma ) avant de rendre un livrable pour vérifier que notre version passe tous les tests.