

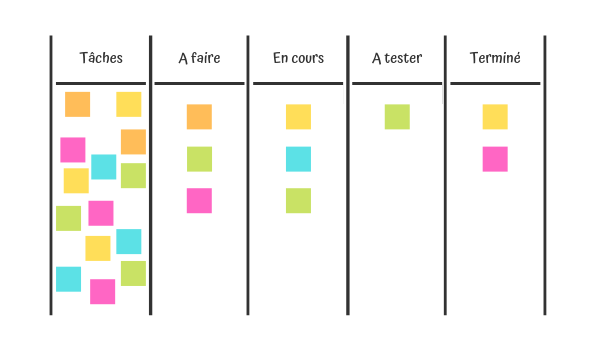
|  |
| --- |
| SIT213 BE - 1  IMT Atlantique |
| Chaire M@D - Maintien à domicile | IMT Atlantique |
| 14 septembre  Créé par :  LE DUC Elouan MAQUIN Philippe LE GRUIEC Clément LE JEUNE Matthieu FRAIGNAC Guillaume |

# 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 cohérent en mise en page utilisation de la même police, même page de garde, etc.

Le contenue doit être accessible pour un lecteur qui n’est pas expert du domaine mais doit être assez précis pour un lecteur expert du domaine.

Les tests sont une partie très importante en effet il nous permette de vérifier le fonctionnement de notre code. Ainsi les tests doivent être assez exhaustif 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 différente personne. Ainsi nous pouvons nous assurer 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 outils de versioning ( Git ). Nous avons choisi GIT pour plusieurs raison GIT nous permet de créer plusieurs branche nous avons décider de créer une branche pour chaque membre de l’équipe dans le quelle il peut travailler. Comme je l’ai dit GIT est un outil de versioning ainsi la fonction reverse nous permet de rattraper des erreurs.

Une fois notre version valide par l’équipe nous pouvons merge notre code dans la branche master qui nous serviras de point de départ pour la prochaine étape du développement.

Pour ne pas que GIT devienne un endroit de non-lieux nous avons imposé certaine règle :

* Des noms explicite pour chaque commit
* Faire des commit régulier mais pas abusif
* Le code doit à minima correctement compiler
* Utiliser des tags ( add et remove ) pour informer des modifications dans les commit

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