

Gestion des risques et rendus

1. Gestion du Planning : répartition des tâches et retards :

Se préparer pour les rendus en se fixant des dates limites pour chaque tâche dans le Gantt.

La gestion des tâches selon leurs ordres de priorité est gérée par le Gantt. Une fois qu'une personne a fini une tâche, il part voir le Gantt et il vérifie les tâches qui ne sont pas encore implémentées. Si quelqu'un se trouve en retard par rapport à une tâche, il est aidé pour pouvoir le débloquer et avancer.

2. Gestion de la Documentation

Documentation : puisque la documentation est très chargée, chacun est spécialisé dans une partie de la documentation, et sur laquelle il est « spécialisé » le plus. Ceci dit, ça ne nous empêche pas de se documenter sur les autres parties mais pas en profondeur. Si quelqu'un a besoin d'une information sur une partie, il consulte la personne en charge qui lui répond où l'envoie vers la partie qui l'intéresse. Grâce à cette solution, chaque membre est capable de réaliser une tâche en entier (cad qui touche les 3 étapes).

3. Gestion des erreurs et organisation :

Une fois une sous-tâche terminée, elle passe à la phase test/vérifications. Un ensemble de tests concernant chaque instruction est testé pour savoir si tout a été bien implémenté et sinon avoir un retour sur la nature du problème afin qu'il soit rectifié au plus vite.

En parallèle, on se documente et travaille sur l'implémentation de l'extension ARM, et ses tests.

Lors de l'implémentation de nos instructions, il est très important de bien commenter son code. Puisqu'il est possible qu'un membre de l'équipe se trouve dans une situation où il doit modifier une partie du code déjà modifié, il peut lire les commentaires et consulter la personne qui a modifié le code.

NB: Toujours compiler pour voir qu'on n'a rien oublié et résoudre nos problèmes avec les outils de débogage.

4. Gestion du Git :

On travaille par branche d'instruction afin de limiter la casse (éviter les conflits) et une fois la tâche achevée, on fait un merge avec le master puis on vérifie la nouvelle version grâce au lanceur de tests automatiques. De cette façon, on est sûr que la branche master dans le git est fonctionnelle sans erreurs de compilation au moins.

5. Gestion des imprévus :

Si par malheur, un ou plusieurs membres n'est pas apte à travailler à cause d'un examen, d'une maladie, un décès dans la famille ou un ordinateur défectueux, il doit informer les membres de l'équipe de son avancement afin qu'il puisse le couvrir. Il doit rester à disposition pour qu'on puisse le consulter si nécessaire. Les commentaires dans les codes sont dans ce cas très utiles.