

Scrum : Projet de maison domotique

Présentation de la méthode scrum :

La méthode scrum est une méthode agile. Avant chaque sprint les tâches sont estimées en temp et complexité de planning poker.

Les rôles de la méthode scrum :

Les méthodes scrum dans notre projet sont définis trois rôles :

- Product owner (prof)
- Scrum master
- Team(l'équipe)

Sprint :

1. Initialisation du sprint :

Les différentes étapes de l'initialisation d'un sprint sous la responsabilité du Scrum master.

2. Release Planning :

Scrum master gère la méthode agile et niveau technique.

Proposer les user story (scrum master et team).

Estimation en point la valeur métier et priorité.

Estimation en temp (développement et intégration)

3. Backlog de produit :

Après avoir le scrum master qui a lancé le projet qui est un projet de maison domotique, il a réuni tout l'équipe pour faire backlog de produit.

Notre équipe à utiliser un « framemo » qui est logiciel gratuit et quand peut faire une sorte de postiches virtuel (voir l'image ci-dessous).



4. Sprint backlog :

Dans cette partie nous avons défini toutes les User story pour ce projet, puis nous avons défini les priorités de chaque User Story (voir GitLab ou ci-dessous dans découpage des tâches).

5. Planning poker :

Il s'agit de noter le point d'effort des user story de 1 à 13 tel que chaque de notre équipe à voter et justifier son choix devant le reste de l'équipe, ce qui permettra pour le reste de l'équipe s'ils l'ont oublié de prendre en compte certains critères quand nous n'avons pas pensé, tel que le scrum master est chargé de noter chaque chiffre pour chaque story (voir l'image ci-dessous)

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|----|--|-------|-------|---------|--------|--------|--------|-------|----------|---------|------------|------|--------|--------|---|---|
| 1 | NOTER DE 1 à 13 (0 1 2 3 5 8 13) | DAMAN | DYLAN | GREGORY | NASSIM | NATHAN | OTMANE | SIHAM | RESULTAT | MOYENNE | Difficulté | | | | | |
| 2 | ROS | | | | | | | | | | | 800 | 25.75 | | | |
| 3 | Installation ROS | 5 | 5 | 1 | 1 | 1 | 3 | 5 | 3 | 8 | | | 62.55 | | | |
| 4 | Configuration ROS | 8 | 8 | 5 | 1 | 2 | 2 | 8 | 4 | | 14 | | 75 | 101.76 | | |
| 5 | Afficher sur l'écran une simulation de mouvement lors de l'ouverture de ma porte principale | 13 | 8 | 8 | 3 | 5 | 8 | 13 | 8 | | | | 150 | | | |
| 6 | Mon système maison réagit à l'ouverture de ma porte principale | 5 | 5 | 5 | 5 | 5 | 5 | 8 | 8 | | | | 112.5 | | | |
| 7 | Je souhaiterais (optionnellement) recevoir le signal à partir du QRCODE qui sera scanné | 13 | 13 | 13 | 13 | 13 | | | 13 | 11 | | | 265.35 | | | |
| 8 | STATUS | | | | | | | | | | | 900 | 25.42 | | | |
| 9 | Avoir en temps réel sur l'interface les données de pression | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 4 | 5 | 4.6 | | 85.89 | 98.84 | | |
| 10 | Avoir instantanément sur l'interface les données de l'état courant de ma porte | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 4 | | | | 85.89 | | | |
| 11 | Afficher sur l'écran une simulation de mouvement lors de l'ouverture de ma porte principale | 8 | 8 | 5 | 5 | 5 | 5 | 8 | 8 | | 3.5 | | 128.52 | | | |
| 12 | Mon système maison réagit à l'ouverture de ma porte principale | 8 | 8 | 5 | 5 | 5 | 5 | 8 | 8 | | | | 128.52 | | | |
| 13 | Connaitre la pression dans mes deux pièces, à partir des cartes STM32 | 5 | 3 | 5 | 5 | 3 | 3 | 5 | 4 | | | | 85.89 | | | |
| 14 | Pouvoir accéder en temps réel sur l'interface à mon ordinateur les données de température de chacune de mes pièces | 8 | 8 | 8 | 8 | 5 | 3 | 8 | 8 | | | | 128.52 | 137.2 | | |
| 15 | Connaitre l'état de ma porte principale (ouvert/fermé), à partir de mon matériel | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | | 8.8 | | 107.1 | | | |
| 16 | Je souhaiterais (optionnellement) recevoir le signal à partir du QRCODE qui sera scanné | 5 | 8 | 8 | 5 | 13 | 8 | 5 | 7 | | | | 149.94 | | | |
| 17 | OPENCV | | | | | | | | | | | 400 | 26 | | | |
| 18 | Scanner un Qrcode avec une caméra | 5 | 3 | 3 | 1 | 2 | 3 | 1 | 2 | 8 | | | 46 | | | |
| 19 | Ouvrir ma porte principale après avoir scanné un Qrcode valide | 13 | 13 | 13 | 3 | 13 | 8 | 5 | 9 | | | | 225 | | | |
| 20 | Intégrer le code openCV dans l'application graphique Qt | 8 | 8 | 8 | 8 | 5 | 5 | 8 | 5 | | | | 120 | 140.9 | | |
| 21 | Qt | | | | | | | | | | | 1100 | 27.24 | | | |
| 22 | Interface graphique intuitive | 8 | 8 | 8 | 5 | 8 | 8 | 13 | 8 | 8 | | | 179.42 | | | |
| 23 | Avoir en temps réel sur l'interface les données de pression | 8 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | | | | 112.2 | | | |
| 24 | | 5 | 3 | 1 | 3 | 3 | 3 | 5 | 3 | | | | 87.32 | | | |
| 25 | Ouvrir ma porte principale après avoir scanné un Qrcode valide | 8 | 13 | 13 | 8 | 13 | 8 | 13 | 10 | | | | 224.4 | | | |
| 26 | Une fenêtre de visualisation des informations de température par pièce | 5 | 5 | 5 | 5 | 5 | 5 | 8 | 5 | | | | 112.2 | | | |
| 27 | Une fenêtre de visualisation des informations de pression par pièce | 5 | 5 | 5 | 5 | 5 | 3 | 5 | 4 | | | | 89.78 | | | |
| 28 | Pouvoir visualiser l'état de la porte principale (ouvert/fermé) | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 2 | | | | 44.88 | | | |
| 29 | Pouvoir accéder en temps réel sur l'interface à mon ordinateur les données de température de chacune de mes pièces | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | | | | 112.2 | | | |
| 30 | Je souhaiterais (optionnellement) recevoir le signal à partir du QRCODE qui sera scanné | 2 | 8 | 5 | 8 | 13 | 5 | 13 | 7 | | | | | | | |
| 31 | | | | | | | | | | | | | | | | |
| 32 | | | | | | | | | | | | | | | | |
| 33 | | | | | | | | | | | | | | | | |
| 34 | | | | | | | | | | | | | | | | |
| 35 | | | | | | | | | | | | | | | | |
| 36 | | | | | | | | | | | | | | | | |
| 37 | | | | | | | | | | | | | | | | |
| 38 | | | | | | | | | | | | | | | | |
| 39 | | | | | | | | | | | | | | | | |
| 40 | | | | | | | | | | | | | | | | |
| 41 | | | | | | | | | | | | | | | | |
| 42 | | | | | | | | | | | | | | | | |
| 43 | | | | | | | | | | | | | | | | |
| 44 | | | | | | | | | | | | | | | | |
| 45 | | | | | | | | | | | | | | | | |
| 46 | | | | | | | | | | | | | | | | |
| 47 | | | | | | | | | | | | | | | | |
| 48 | | | | | | | | | | | | | | | | |
| 49 | | | | | | | | | | | | | | | | |
| 50 | | | | | | | | | | | | | | | | |



6.Découpage des tâches :

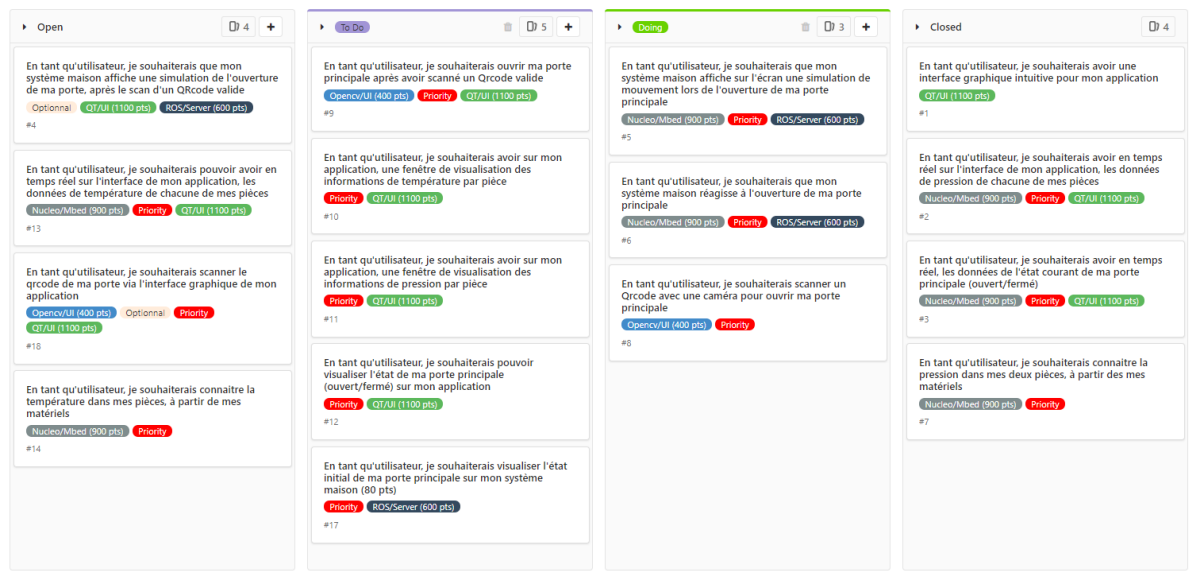
Le découpage en tâches est l'étape qui survient juste après planning poker

○ Les sous tâches :

Nous nous sommes divisés les User Story en sous-tâches pour rendre le travail plus simple en fonction des compétences de chacun ensuite le scrum master se charge en suite de saisir les User story pour chacune des sous taches dans gitlab

Le scrum board est divisé en quatre colonnes qui est présenté sur gitlab (voir l'image ci-dessous) :

- Les tâches ouvert (qui ne sont pas encore traiter).
- Les tâches à faire.
- Les tâches en cours.
- Les tâches terminées.



Après avoir fait le backlog, planning poker et découpage des tâches nous allons commencer le projet tels que :

7.Daily Scrum :

Nous avons utilisé daily scrum chaque matin 15 min avant de commencer le projet tels que le scrum master pose les questions suivantes pour chacun de nous :

- Qu'est ce vous avez fait hier ?
- Quelles sont les difficultés rencontrées ?
- Qu'est ce vous allez faire aujourd'hui ?

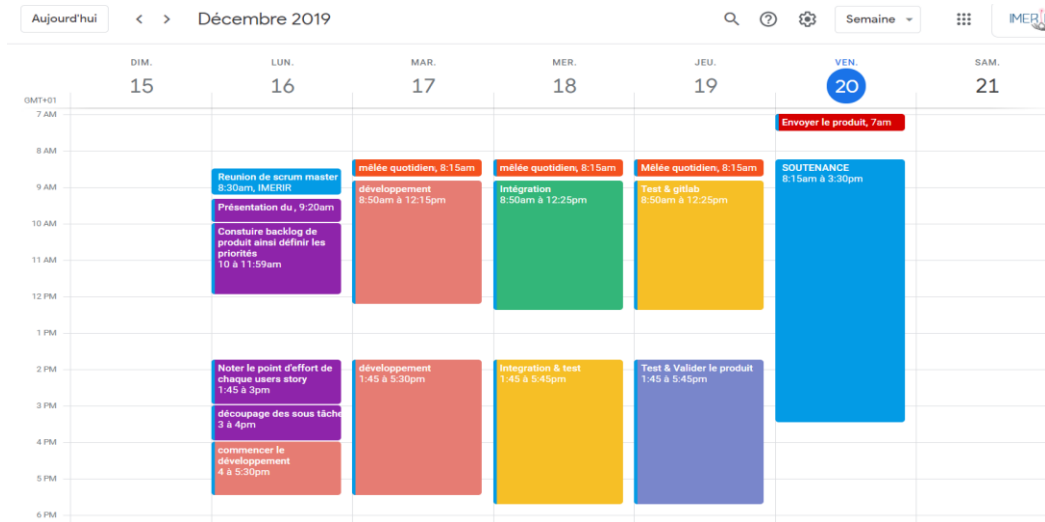
Puis pour les difficultés rencontrées nous allons discuter tous ensemble soit une solution proposée par l'équipé ou proposer de l'aide

8.Revue de sprint :

Pour cette revue, nous avons récapitulé les User Story complétées, les difficultés rencontrées pendant le Sprint ce qui est à améliorer et qu'est-ce que nous allons faire pour le prochain Sprint.

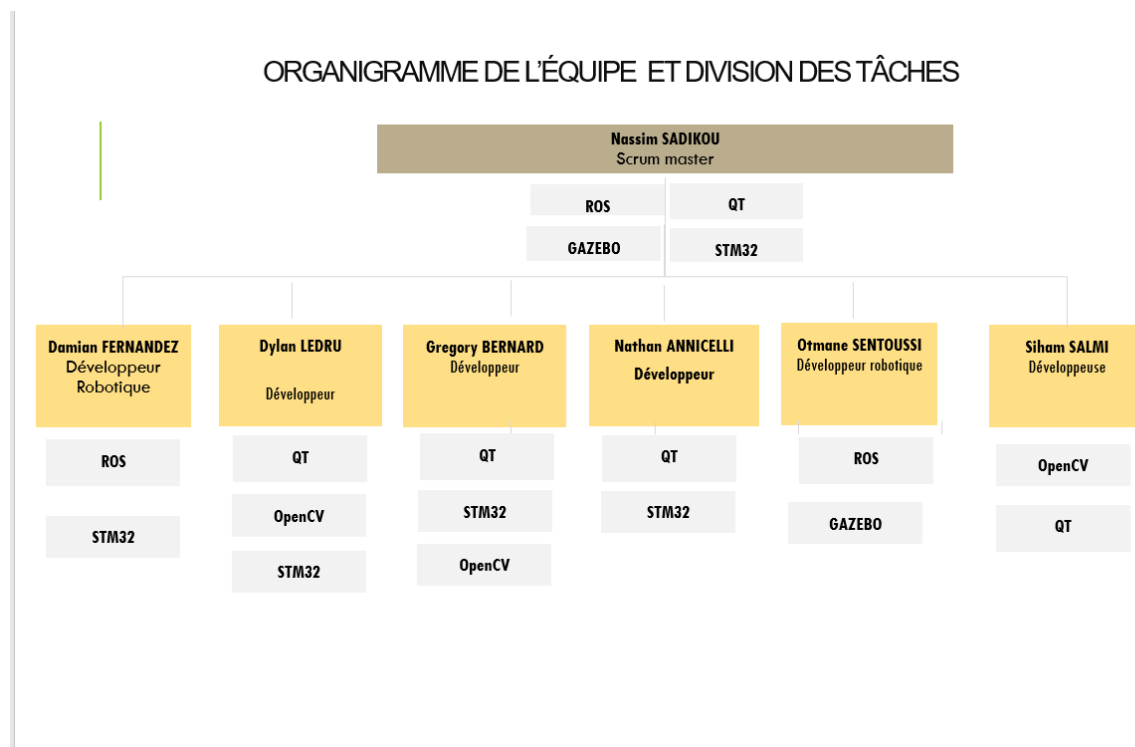
9. Rétrospective de sprint :

Nous avons parlé de l'organisation du sprint, l'organisation du travail ainsi que notre intégration dans l'équipe.



10. Répartition des tâches :

Présentation l'organigramme des tâches de chaque personne de notre équipe.



11. L'explication les parties :

- **ROS :**
 - Installation et configuration ROS avec des commandes de linux.
 - Préparation du script python qui permet la communication entre le python et la STM32.
 - Initialisation de l'espace 3d sur Gazebo pour la simulation de l'ouverture de la porte.
 - Adaptation du script python pour mettre la publication sur Gazebo.
 - Automatisation de toute la procédure avec un fichier. launch.
- **QT :**
 - Partage des tâches, une partie pour s'occuper de l'interface graphique, et une autre pour récupérer et traiter les données reçues par les cartes STM32. Ce partage a été choisi pour être plus efficace dans le développement Qt et permettre une intégration rapide du code.

- **STM32 :**
 - Compilation à l'aide du compilateur Mbed en ligne pour faciliter la programmation et donc nous permettre de gagner du temps.
 - Récupération des codes sur la page Nucléo de notre matériel car les codes Example existaient déjà.
- **OPENCV :**
 - Récupération du code de détection QR code sur internet pour gagner du temps sur la programmation.
 - Adapter le code de détection QR code à notre projet tels qu'il affiche sur le terminal la porte est ouverte ou fermé c'est QR code valide ou pas.
 - Intégration au QT pour faciliter la communication des états de porte (ouvert/fermé).
 - Mise en place d'un thread à part pour OpenCV pour éviter l'interférence avec la lecture des cartes Nucléo.

12. Test le produit :

- **Test d'intégration :**

Des tests permettant de tester la bonne intégration de la fonction dans l'ensemble du logiciel et de ses dépendances.

Parmi les tests quand nous avons fait c'est :

 - Test (STM32 et QT)
 - Test (openCV et QT)
 - Test (STM32 et ROS)
 - Test (STM32, OpenCV et QT)
 - Test (gazebo et ROS)
- **Test de Qualité :**

Il s'agit ici de vérifier la qualité du code à l'aide d'un certain critère :

 - Fonctions inutilisées ;
 - Constantes inutilisées ;
 - Taux de commentaires dans le code ;
 - Redondance dans les fonctions ;
 - Constantes définies plusieurs fois ;
 - Warning ;

13. Validation, incrément de produit livrable :

- **Validation :**

Une fois les user-story d'un sprint réalisé puis livrées, l'ensemble du sprint fait l'objet d'une livraison au client. Celui-ci, au bout d'une ou plusieurs livraisons d'itérations dans notre cas c'est nous qui se mette à la place d'utilisateur

- **Produit livrable :**

Après les tests valide, le projet réussi avec la volonté de l'équipe et l'utilisation de la méthode scrum ainsi respecter le délai de projet, le projet prêt à être livrer