

Mycélium 3.0 - Planification

CHEREL Valentin - DERRIEN Thomas - LE GOASTELLER Maïwenn
LESSIRARD Leo - MAUGER Arthur - OUVRARD Pierre - POURCHER Pierrick

Encadrants : PARLAVANTZAS Nikos - PAROL-GUARINO Volodia

Avec la participation de :
MOUREAU Julien

2023-2024



FIGURE 1 – Mycélium : projet de suivi environnemental

Table des matières

1	Introduction	3
2	Les entrants (contexte)	3
2.1	Acteurs	3
2.2	Périmètre fonctionnel	3
2.3	Éléments en entrée	4
2.4	Périmètre de qualification	4
2.5	Calendrier	5
2.6	Pilotage	5
3	Analyse de risque	5
3.1	Identification des risques	5
3.1.1	Risques projet	6
3.1.2	Risques produits	6
3.2	Classer les risques	6
3.3	Plan d'action	7
4	Organisation	8
4.1	Cycle de production	8
4.1.1	La méthode (agile, V)	8
4.1.2	Les Phases et jalons (dates clés)	8
4.1.3	Les productions (documents, exécutable...)	8
4.2	Cycle de qualification	9
4.2.1	Quand (Tests unitaires, intégration continue)	9
4.2.2	Comment (Simulateurs, jeux de test)	9
4.3	Organisation	9
4.4	Mode de pilotage du projet	10
5	Estimation des tâches	10
5.1	Rappel des hypothèses	10
5.2	Découpage	10
5.3	Structuration de l'estimation en WBS	10
6	Planification	11
6.1	Hierarchie des tâches	11
6.2	Affectation des ressources par tâche	12
6.3	Planning	12
6.4	Graphe d'occupation des ressources	12
7	Conclusion	13
8	Annexes	14

1 Introduction

2 Les entrants (contexte)

2.1 Acteurs

Différents partis interagissent avec notre projet, la clientèle, l’encadrement et les membres de l’équipe projet.

Clients

Laurent LONGUEVERGNE (Directeur de recherche à l’Observatoire des Sciences et de l’Univers de Rennes) et Julien MOUREAU (Ingénieur d’étude pour l’Institut de Physique du Globe de Paris en mission à l’Université de Rennes 1 avec Laurent LONGUEVERGNE) assurent le rôle de clients. Ils participent régulièrement à des réunions avec l’équipe projet dans lesquelles ils expriment notamment leurs besoins et retours.

Encadrants

Nikos PARLAVANTZAS (enseignant-chercheur à l’INSA Rennes et à l’IRISA dans l’équipe Myriads) et Volodia PAROL-GUARINO (Doctorant à l’INRIA) constituent l’encadrement de ce projet.

Equipe projet

Valentin CHEREL, Pierrick POURCHER, Thomas DERRIEN, Leo LESSIRARD, Pierre OUVRARD, Arthur MAUGER, Maïwenn LE GOASTELLER, tous étudiants l’ingénierie informatique à l’INSA Rennes.

2.2 Périmètre fonctionnel

La construction de la ligne B du métro a généré des dommages sur l’écosystème. Afin de compenser ces dommages et de préserver l’écosystème local, la municipalité de Rennes a pris l’engagement de lancer des projets de renaturation visant à aménager environ 35 hectares au sein même de la ville.

C’est dans ce contexte que le projet Mycélium a vu le jour en 2021, portant un objectif clairement défini : avoir un suivi de la renaturation de La Croix Verte, un espace végétalisé situé au cœur du campus de Beaulieu. Plus globalement, ce travail s’inscrit dans le projet de recherche **TERRA FORMA**, qui a pour objectif de maintenir constamment à jour une évaluation de nos ressources naturelles, telles que le sol, l’eau et la biodiversité.

Ce projet vise à équiper cette zone de capteurs intelligents pour collecter des données cruciales à propos de la réaction de l’écosystème aux changements environnementaux et aux contraintes climatiques. Il se charge également de la récupération et l’analyse de ces données.

L’objectif de notre équipe pour cette année est de s’approprier le projet et de l’améliorer. Nous nous concentrerons en particulier sur la simplification de l’architecture de celui-ci, la simplification de sa compréhension ainsi que la simplification de sa prise en main. Pour cela, nous allons analyser différents composants du projet et voir s’ils sont superflus ou si l’on peut les améliorer en remplaçant la technologie utilisée. Une autre grande partie du projet sera dédiée à faciliter la prise en main du projet. Cela passe par la réalisation de guides détaillés permettant d’aiguiller l’installation et la prise en mains des technologies. Ainsi que la mise en place d’un **cluster de**

Machines Virtuelles (VMs). Ce cluster permet de simuler le matériel réel dont nous disposons (un cluster de cinq Raspberry Pi). Il permet de réaliser de nombreux tests sans avoir de conséquence sur le matériel, et sans avoir à se préoccuper d'éventuels problèmes causés par le matériel. Ce cluster sera ensuite plus simple à léguer aux équipes suivantes.

2.3 Éléments en entrée

Ayant pour but d'être déployé dans des environnements divers, notamment ruraux, les contraintes associées au projet sont assez nombreuses. En effet, Mycélium peut être déployé dans des milieux qui peuvent être isolés et nécessiter différents types de capteurs. Il nous faut donc proposer une solution assez générale qui permette de suivre différents climats et événements. De plus, le lieu d'installation des capteurs peut être éloigné du lieu de traitement des informations, ce qui pose des contraintes pour le transit de ces données via **LoRaWAN**.

Le premier groupe s'est concentré sur la création de scénarios, qui sont des événements pouvant être observés à l'aide des capteurs. Dans cette première version du projet, sept ont été implémentés.

Les membres du groupe ont utilisé les fonctionnalités du capteur à leur disposition pour prendre des mesures à une certaine fréquence, les enregistrer, et envoyer des données sur le cloud. De plus, ils ont ajouté trois nouvelles fonctions pour ces scénarios. Deux de ces fonctions sont destinées à traiter certains des scénarios mentionnés précédemment, en déclenchant des actions lorsque certains seuils sont atteints. La troisième fonction sert à récupérer des informations au démarrage des capteurs.

Ils ont également déployé l'infrastructure du cluster Mycélium en utilisant plusieurs Raspberry PI pour traiter les données envoyées par le **nœud SoLo** via le protocole LoRaWAN. Ils ont également créé des fonctions **OpenFaaS** pour opérer directement sur ces données. Enfin, ils ont mis en place des fonctionnalités liées à la visualisation et au traitement des données, notamment l'envoi de notifications en cas d'événements climatiques, des tests de qualité, la comparaison avec l'historique des données et le stockage dans une base de données **InfluxDB**.

Le deuxième groupe a continué le projet, en consacrant d'abord un temps à l'appropriation de celui-ci. Ils ont créé trois nouveaux scénarios :

- Le scénario "Normales saisonnières" qui notifie lorsque les températures détectées sortent des normales saisonnières.
- Le scénario "Orage" qui notifie la présence d'épisodes orageux.
- Le scénario "FoxyFind" qui notifie la présence d'animaux devant une caméra.

Les ressources du cluster disponibles étant moindres et rapidement consommées lors du fonctionnement, ils ont entrepris l'ajout d'un lien avec le cloud dans le but de déléguer une partie des calculs afin d'éviter la surcharge. Cependant, seules les fonctionnalités liées au scénario FoxyFind étaient fonctionnelles sur le cloud lors du projet, ce qui n'est pas totalement satisfaisant.

2.4 Périmètre de qualification

Notre projet a pour objectif principal de créer un système permettant de réaliser des relevés de données environnementales, et de traiter celles-ci. Il fait suite aux projets Mycélium 1.0 et 2.0 des groupes de projet de 4e années des deux années

précédentes. Plus précisément, nos objectifs sont de mettre en place la capture des données, d'établir la communication entre les capteurs et le cluster de Raspberry Pi (ou la machine virtuelle émulant le cluster), ainsi que de permettre le traitement des données et le monitoring du système.

Pour ce faire, nous disposons des technologies et du matériel suivant :

- **Cluster de Raspberry PI** : Un ensemble de Raspberry PI qui permet de réaliser de premiers traitements avec une faible consommation d'énergie.
- **Réseau LoRa** : Réseau à faible consommation et faible débit qui permet de communiquer avec les capteurs sur de grandes distances.
- **VM** : Machine virtuelle permettant d'émuler le comportement du cluster de Raspberry Pi afin de travailler sans nécessiter le cluster physique.
- **VPS** : Serveur privé virtuel hébergé par la DSI de l'INSA nous permettant de déléguer une partie des calculs nécessitant plus de puissance de calcul que ce que permettent de supporter les Raspberry PI.

2.5 Calendrier

Nous disposons d'une équipe de 7 étudiants et étudiantes ingénieurs informaticiens en 4e année d'étude à l'INSA. Nous commençons le projet à 100% de nos capacités pendant le 1er semestre de septembre à janvier. À partir de janvier, pendant le 2d semestre, 2 membres de notre équipe partent en mobilité universitaire à l'étranger, ils cessent en conséquence de travailler sur le projet Mycélium, nous ne serons donc plus que 5 dans l'équipe, soit 71,4% de nos capacités (hypothèse où les capacités sont également réparties pour tous).

Le projet donne lieu à diverses livraisons réparties au fur et à mesure de l'année :

- Rapport de spécification (24/11/23)
- Rapport de planification (15/12/23)
- Soutenance de planification (21/12/23)
- Soutenance de projet (22/12/23)
- Rapport de conception (15/02/24)
- Page web (28/03/24)
- Rapport final (02/05/24)
- Présentation du projet aux 3INFO (06/05/24)
- Livraison finale (06/05/24)
- Soutenance finale (07/05/24)

2.6 Pilotage

Concernant le pilotage du projet, nous avons une réunion par semaine à un horaire fixe, le lundi de 15 h 45 à 17 h 45, cela nous permet de discuter de l'avancement du projet avec l'encadrement du projet, et de pouvoir faire un point avec les membres de l'équipe.

3 Analyse de risque

3.1 Identification des risques

Nous allons dans cette partie aborder les différents risques liés à Mycélium 3.0.

3.1.1 Risques projet

Commençons par les risques qui peuvent provoquer un dysfonctionnement du projet.

Le premier risque est en lien avec la modification et la suppression de différentes technologies de notre projet. En effet, dans le but d'éviter des pertes de ressources actuellement présentes, nous avons décidé de supprimer des technologies inutiles et de remplacer certaines technologies par d'autres qui nécessitent moins de ressources. Nous pouvons donc avoir ici un risque de **régression** du projet, dans le cas où au final ces nouvelles technologies sont moins performantes que les anciennes.

Un autre risque est lié à la **compatibilité**. Pour notre projet, nous allons développer non pas sur le support d'origine, mais sur des machines virtuelles afin de faciliter le développement. Cependant, il y a le risque que lorsque l'on déploie ce que l'on a développé sur le support d'origine, on se retrouve face à une incompatibilité.

Un autre risque est lié à la **performance** de Mycélium 3.0. C'était déjà un problème rencontré par les anciennes versions de Mycélium. Le problème est que le système sur lequel se base le projet est très peu performant, surtout au niveau de la mémoire RAM ce qui amène rapidement à sa saturation.

Il y a également le risque d'une **perte d'accès** à différentes données. Pour notre projet, nous récupérons des données, soit à partir de capteurs, soit directement dans une base de données de l'**OSUR** (Observatoire des sciences de l'univers de Rennes). Or il y a un risque de dégradation (à cause de la météo) ou de vol pour les capteurs. Concernant la base de données de l'OSUR, si celle-ci est désactivée pour une raison indépendante de notre volonté, nous perdrons également l'accès à ces valeurs.

3.1.2 Risques produits

Concentrons nous à présent sur les risques qui peuvent provoquer un mauvais déroulement du projet.

Le premier risque pouvant impacter le bon déroulement du projet est lié au **manque de compétences initiales** des membres du projet. Peu de membres du projet maîtrisant les technologies initiales, il nous faut nous les assimiler et cela nécessite un certain temps.

Nous avons aussi le **périmètre du projet**. En effet, ce dernier était initialement très grand, avec beaucoup de possibilités. Il y a donc un risque que les choix que nous avons décidé de faire pour ce projet nécessitent plus de temps que prévu, le départ de deux membres du groupe au cours du projet pouvant nous impacter négativement.

3.2 Classer les risques

Nous allons à présent classer les risques et étudier leur probabilité d'apparition et leur répercussion sur le projet, comme indiqué dans la table 1 ci-dessous.

Risque	Probabilité	Répercussion
Manque de Performance	Forte	Forte
Incompatibilité	Moyenne	Forte
Régression	Moyenne	Moyenne
Perte d'accès aux données	Faible	Moyenne
Périmètre trop grand	Moyen	Faible
Manque de compétences	Forte	Faible

TABLE 1 – Matrice d'analyse des risques

Le risque le plus important est celui du manque de performance du cluster. En effet, c'était un problème connu dès les précédentes années donc il y a grande chance que nous le rencontrons et que ça empêche le bon fonctionnement du projet.

Viennent ensuite les risques de compatibilité. Nous avons mis une probabilité d'occurrence moyenne, car c'est un cas assez incertain. Ce que nous savons, c'est que si nous le rencontrons, les répercussions sur le projet seront très grandes, car tout notre projet sur machine virtuelle sera inutilisable sur le cluster et nécessitera une grande charge de travail pour le rendre compatible.

Les risques de régression et d'un périmètre trop grand ont une probabilité moyenne. La régression aura un impact assez important, car il faudra dans ce cas revenir à l'ancienne technologie et cela nécessitera une charge de travail importante. Pour le périmètre les répercussions sont moindres, car tant qu'on réalise dans un premier temps les tâches essentielles au fonctionnement du système, dans un second temps ce seront les tâches secondaires qui pourront soit être annulées soit être transférées à une prochaine version de Mycélium.

La perte d'accès aux données à peu de chance d'arrivée, et si ça arrive les conséquences seront importantes, mais pas catastrophiques, car on aura toujours moyen de trouver d'autres données.

Pour finir, le manque de compétences à une probabilité très forte, mais des répercussions faibles, car il nous suffira de nous autoformer pour que le projet se déroule correctement. Cependant, si cette autoformation prend trop de temps les répercussions risquent d'être bien plus importantes, car ça pourrait jouer sur le bon maintien des délais.

3.3 Plan d'action

Nous avons vu dans la partie précédente que certains risques peuvent avoir de grandes répercussions sur le projet. Nous allons voir dans cette partie comment nous allons gérer ces risques pour éviter de les rencontrer.

Commençons par le risque de compatibilité. Pour diminuer le risque d'occurrence, nous allons réaliser nos machines virtuelles de façon qu'elles correspondent le plus exactement possible à la configuration du système d'origine. Ainsi, on diminue de manière conséquente le risque d'incompatibilité.

Ensuite pour le manque de performance, nous avons décidé de poursuivre l'idée du projet Mycélium 2.0 et d'utiliser un VPS, qui est une machine virtuelle stockée sur un serveur de l'INSA. Ainsi, le système d'origine pourra recourir à ce VPS afin de diminuer sa charge de travail et ainsi éviter une saturation de la mémoire du cluster. De cette façon, on évite le manque de performance de notre projet. Cependant, on ajoute un risque de dysfonctionnement dans le cas où les serveurs de l'INSA sont hors service.

Pour éviter la régression, nous allons avant de supprimer une technologie faire une étude pour confirmer son inutilité. Idem pour les modifications de technologies, où nous ferons des évaluations des performances des deux technologies afin d'être sûrs que le changement est judicieux. Même si cela ne supprime pas le risque de régression, cela le diminue de manière conséquente.

Pour finir, même si ce n'était pas le risque le plus important, nous avons décidé de trouver une solution au manque de compétences. Pour cela, dès qu'une personne maîtrise une technologie, un guide est réalisé. Ce guide servira à la fois aux autres membres du projet à prendre en main la technologie sans perte de temps, mais il servira également aux futurs membres des prochaines versions de Mycélium à prendre le projet en main sans perdre autant de temps que nous.

4 Organisation

4.1 Cycle de production

4.1.1 La méthode (agile, V)

Nous adoptons une approche hybride dans notre méthodologie de développement. Pour les aspects techniques ainsi que pour l'écriture du guide, nous préférons adopter une approche Agile, afin de favoriser la flexibilité et l'adaptabilité tout au long du processus. En ce qui concerne les rapports à rendre, nous réalisons des cycles en V, bien que ces principes ne soient pas strictement suivis de manière rigide.

4.1.2 Les Phases et jalons (dates clés)

Notre cycle de production a débuté par une phase de conception et de réflexion approfondie concernant l'architecture du projet. Cette phase est ensuite suivie par une période de développement, où des tests sont menés régulièrement tout au long du processus. Cependant, nous restons ouverts aux ajustements et aux changements nécessaires qui pourraient émerger pendant le développement, notamment concernant les améliorations que nous souhaitons apporter. Ces changements pourraient entraîner un retour vers une phase de conception itérative. Ce processus itératif se poursuit jusqu'à ce que nous envisagions le déploiement final de notre architecture.

4.1.3 Les productions (documents, exécutables...)

Pendant la durée du projet, nous allons produire divers rapports. La liste des rapports ainsi que leurs différentes dates de rendus peuvent être retrouvées dans la section 1.5 du rapport. Tous ces rapports sont stockés et accessibles à l'ensemble de l'équipe via un espace partagé sur Google Drive. En ce qui concerne le code que nous allons produire, nous l'hébergerons sur un dépôt GitLab dédiée au projet. Contrairement à certaines pratiques antérieures, nous voulons rassembler l'ensemble

du code source dans un seul dépôt GitLab afin d'éviter toute confusion et de simplifier la gestion du code. Par ailleurs, les rapports seront également accessibles sur cette plateforme pour garantir une documentation complète et centralisée du projet. C'est aussi sur cette plateforme que nous mettrons à disposition les différents guides que nous allons créer.

4.2 Cycle de qualification

4.2.1 Quand (Tests unitaires, intégration continue)

Au cours de chaque étape du développement de l'architecture, nous avons prévu d'effectuer des tests pour évaluer le bon fonctionnement des nouveaux ajouts. Bien que nous n'ayons pas encore confirmé la mise en place de tests unitaires, cette possibilité reste envisagée. Cette décision dépendra de la faisabilité technique ainsi que des contraintes au niveau du temps dont nous disposons.

4.2.2 Comment (Simulateurs, jeux de test)

Pour la réalisation de nos tests, nous comptons principalement sur des tests manuels. Par exemple, pour vérifier la réception de données dans le cluster, nous prévoyons d'utiliser un capteur disponible pour envoyer des données et ainsi valider la réception et le traitement correct dans le système. Ces approches nous permettront d'évaluer la robustesse et le bon fonctionnement de l'architecture en cours de développement, tout en restant flexibles pour éventuellement intégrer des tests unitaires si cela s'avère réalisable et bénéfique dans le cadre du projet.

4.3 Organisation

Notre organisation repose sur une structure en groupes thématiques flexibles, chacun dédié à des tâches spécifiques au sein du projet. Ces groupes seront expliqués plus en détail dans la partie 4.2 de ce rapport. Pour faciliter la communication et la collaboration au sein des différents membres du groupe et l'encadrement du projet, nous utilisons divers outils numériques. Principalement, nous utilisons un serveur Discord dédié, qui nous permet d'échanger rapidement et facilement, que ce soit depuis un ordinateur ou un téléphone. Ce canal de communication instantanée garantit une connectivité rapide entre tous les différents membres du groupe. En ce qui concerne le partage de documents et de ressources, nous utilisons Google Drive, offrant ainsi une plateforme collaborative pour le stockage et la création de documents. Cette solution présente plusieurs avantages significatifs, notamment la possibilité de travailler simultanément sur un même document, la sauvegarde automatique des modifications, la capacité à ajouter des commentaires et un chat en temps réel. Pour le partage et la gestion du code source, nous utilisons GitLab, un système dédié qui facilite la collaboration entre les membres de l'équipe tout en assurant un suivi précis des modifications apportées au code. Enfin, pour la création des différents rapports utilisant LaTeX, nous utilisons la plateforme en ligne Overleaf. Cet outil nous permet de travailler de manière collaborative sur un seul document, de compiler directement le contenu et de suivre l'évolution du rapport en temps réel, avec un historique disponible en cas de besoin.

4.4 Mode de pilotage du projet

Pour ce projet, nous allons partir sur une organisation où la fonction de responsable de projet sera assurée à tour de rôle par les membres du groupe. Le changement se fera tous les premiers lundi du mois. Cette fonction aura comme responsabilités la bonne tenue du planning, de réfléchir au contenu des différents sprints avec l'aide des autres membres, de s'assurer que tout le monde avance sans problème et du suivi du projet sur Microsoft Project.

Nous sommes partis sur un rythme de 12h par semaine : 3h le lundi et jeudi et 2h le mardi, mercredi et jeudi.

5 Estimation des tâches

5.1 Rappel des hypothèses

L'équipe Mycélium 3.0 est composée de 7 membres durant le Semestre 7, et sera composée de 5 membres durant le Semestre 8, suite aux départs en mobilités de deux membres du groupe. Nous avons décidé de séparer l'équipe en deux parties pour le début du projet. Une équipe se charge de la partie capture et réception des données, tandis que l'autre équipe se charge de la partie traitement de données.

La première équipe se compose de Thomas DERRIEN, Léo LESSIRAD et Arthur MAUGER. Le matériel utilisé comprend un nœud SoLo ainsi qu'une gateway. Les principales technologies importantes pour cette partie sont le réseau LoRaWAN ainsi que le logiciel Chirpstack.

La deuxième équipe se compose de Valentin CHEREL, Maiwenn LE GOASTELLER, Pierre Ouvrard et Pierrick POURCHER. Le matériel utilisé est le cluster de Raspberry Pi, même si au début tout se fera sur un émulateur. Les principales technologies utilisées par cette équipe sont Kubernetes, MQTT et NATS.

Ces équipes sont vouées à être modifiées au cours du projet, pour s'adapter aux tâches en cours et rester le plus efficace possible. En effet, si une des équipes termine ses tâches, ou si une équipe est en difficulté, il convient de réorganiser la répartition du travail au sein des membres du projet.

5.2 Découpage

Nous avons listé les différentes tâches que nous devons réaliser lors du projet afin d'accomplir tous les objectifs que nous nous sommes fixés. Ces tâches ont été classées de façon thématique, par rapport à la fonction au sein de l'architecture : Capture des données, envoi, traitement puis retour utilisateur. Il y a aussi des tâches liées aux rendus que l'on doit remettre au cours du projet.

5.3 Structuration de l'estimation en WBS

Pour chacune des tâches évoquées dans la partie précédente, nous avons associé un temps en heure. Ce temps correspond à une estimation du nombre d'heures que la tâche doit prendre. Ces durées ne sont pas individuelles, mais correspondent à la somme des durées que va passer chaque membre du groupe sur cette tâche. Bien entendu, il ne s'agit là que d'estimations, car en cas de problèmes spécifiques que l'on peut rencontrer ou non sur les tâches, il se peut que cela impacte grandement

la durée de celles-ci. Si une tâche s'avère plus simple que prévu, nous pourrions alors consacrer le temps gagné pour nous attarder sur les tâches plus ardues.

Afin d'effectuer toutes ces tâches, nous avons prévu de travailler 12h par semaine et par personne. S'il s'avère que nous n'avons pas le temps d'effectuer toutes les tâches, certaines tâches ne seront tout simplement pas effectuées. Au sein des différentes catégories de tâches, nous les avons classées par ordre d'importance, et donc par ordre chronologique, car nous allons effectuer en priorité les tâches les plus importantes.

Nous pouvons trouver l'ensemble des tâches ainsi que la durée estimée de ces tâches dans la table n°2.

6 Planification

6.1 Hiérarchie des tâches

Cette section va reprendre le découpage des tâches en 6 parties regroupées dans le tableau vu précédemment (Table 2).

La première partie est le travail préliminaire que l'on a effectué précédemment, elle nous a permis de découvrir le projet pour le comprendre, d'étudier les technologies utilisées et de commencer à mettre en place le projet. Il était important de commencer à toucher aux différentes technologies du projet pour nous rendre plus facilement compte des difficultés que l'on pourra rencontrer et permettre une meilleure planification. Elle est conclue par la remise d'un rapport de spécification du projet.

La capture des données est une partie primordiale pour la suite du projet puisqu'en effet, pour que le projet fonctionne, il faut des données récoltées sur le terrain. Il faut alors travailler sur la récolte de ces données.

Cela nous mène à une partie nécessaire au projet qui est la communication entre tous les capteurs qui capturent les données et notre cluster qui permet de recevoir et de traiter ces données. Nous devons alors travailler sur la communication et la réception.

Après avoir reçu les données, il faut les traiter. C'est sur cette partie que nous allons essayer de simplifier et d'alléger au maximum le travail fait sur ce projet les années passées, ce qui est notre objectif principal.

Par la suite, il faut que les utilisateurs et utilisatrices et les futurs étudiantes et étudiants qui prendront en main le projet puissent facilement utiliser le dispositif et comprendre comment l'installer, le configurer. C'est pourquoi dans cette partie, nous ferons principalement un guide pour les futurs usagers. Ce guide sera écrit au fur et à mesure du projet pour avoir tous les détails et les difficultés qui peuvent être rencontrées.

La dernière partie contient l'écriture des différents rapports et rendus que l'on doit réaliser, ainsi que la préparation aux soutenances qui seront faites tout le long du projet puisque la documentation et la présentation du travail sont obligatoires pour les projets 4INFO. C'est une partie qui prend beaucoup de temps et qui sera mise au premier plan à l'approche des différents rendus qui rythment le projet.

6.2 Affectation des ressources par tâche

Nous allons maintenant voir la répartition des tâches entre les différentes ressources. Jusqu'à présent le travail a été réparti entre les 7 membres de l'équipe, mais en raison du départ en mobilité académique de 2 membres du groupe, à partir de la rentrée 2024 l'équipe ne sera plus composée que de 5 personnes qui se concentreront sur la partie plus technique du projet.

Les différentes tâches seront alors réparties entre les ressources restantes en priorisant les tâches les plus importantes déjà définies pour que le projet soit utilisable à la fin même si des difficultés étaient rencontrées.

Deux groupes pourront se former, l'un traitant la capture et la communication et le second faisant le traitement des données. En effet, le reste des tâches est principalement à faire en parallèle.

6.3 Planning

Les activités, découpées et quantifiées, ont été intégrées dans un projet Microsoft Project. Une fois le projet configuré en tenant compte des périodes de travail, le nivellement automatique peut être réalisé. Les frises chronologiques résultent de cette planification (voir figures 5a, 5b, 5c, 5d), offrant ainsi la possibilité d'établir un calendrier pour les équipes.

La figure 5a est une vue globale du projet, le reste des figures sont des vues plus détaillées des différentes parties du projet. Les parties sur la documentation et le lien avec l'utilisateur n'ont pas besoin d'être détaillés puisque ce sont principalement des tâches dispersées faites en parallèle tout le long du projet.

6.4 Graphe d'occupation des ressources

Le projet étant conséquent, des priorités ont été mises sur les tâches pour faire d'abord les plus importantes. En effet, nous sommes conscients que selon les difficultés rencontrées, nous pourrions ne pas finir toutes ces tâches. Cela se retranscrit dans le graphe d'occupation des ressources puisque même si des périodes moins denses sont présentes, il y a aussi beaucoup de périodes où le travail sera important. C'est pourquoi nous ferons d'abord les tâches les plus prioritaires pour accomplir les principaux objectifs du projet. En revanche, même si les ressources sont en rouge, si on regarde dans les détails sur Microsoft Project, le nombre d'heures de travail de l'équipe n'est pas dépassé par rapport aux horaires prévus. Ces dépassements apparaissent lorsque plusieurs tâches se font la même journée même si elles ne se font pas en même temps. Finalement, les ressources ne sont pas en surutilisation même si le graphe en montre une.

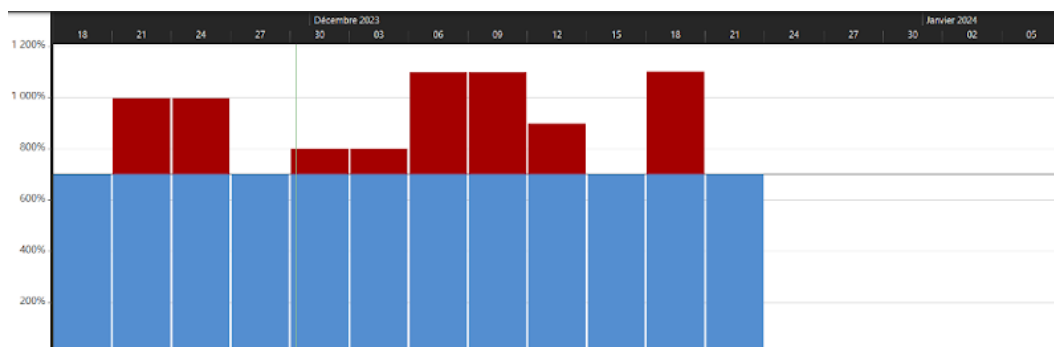


FIGURE 2 – Graphe d'occupation des ressources 1ère partie

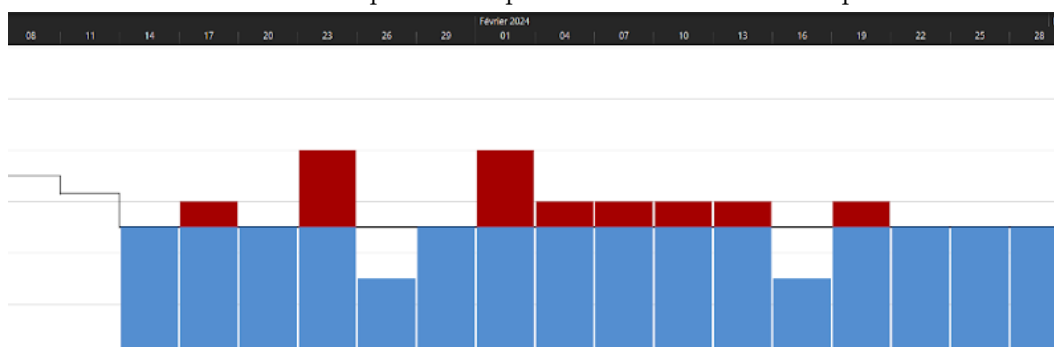


FIGURE 3 – Graphe d'occupation des ressources 2ème partie

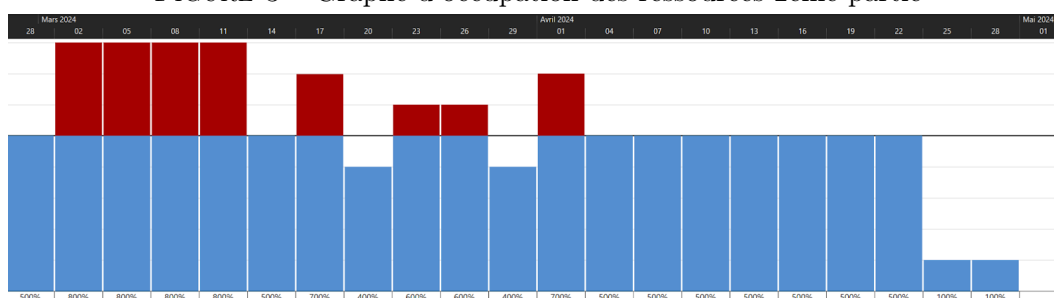


FIGURE 4 – Graphe d'occupation des ressources 3ème partie

7 Conclusion

La réalisation du projet Mycélium 3.0 est un défi complexe, abordant divers aspects techniques et organisationnels. Son objectif principal de surveillance de la renaturation de La Croix Verte à Rennes suite à la construction du métro est associé à un ensemble de risques et de contraintes. Cependant, ces ambitions sont accompagnées de multiples risques, notamment liés à la complexité technologique, aux compétences initiales des membres de l'équipe et aux futurs départs de membres.

Pour répondre à ces défis, l'équipe a adopté une approche hybride dans sa méthodologie de développement, combinant des éléments Agile et des éléments tirés du modèle en V pour assurer à la fois la flexibilité et la rigueur nécessaire au projet. Les phases de production, de qualification et d'organisation ont été réfléchies pour garantir une progression cohérente du travail, tout en tenant compte des contraintes de temps et des spécificités de chaque tâche. Une structuration claire des tâches a été définie, avec des estimations de temps ainsi que des priorités.

Nous avons aussi réalisé des frises chronologiques spécifiques pour chaque phase du projet, offrant ainsi une vision temporelle globale du projet. De plus, les graphes d'occupation des ressources mettent en évidence la nécessité de prioriser certaines tâches pour assurer le bon déroulement du projet.

8 Annexes

Tâches	Durée estimée (heures)
Travail préliminaire (avant spécification)	200
Mise en place de la VM	50
Appropriation des différentes technologies	70
Rapport Spécification	80
Capture des données	205
Mise en place de la capture des données	40
Système d'identification des capteurs	25
Système de localisation des capteurs	15
Simplification de l'ajout de nouveaux capteurs	75
Rajout du capteur FoxyFind	50
Communication	155
Communication capteurs-gateway-Chirpstack	75
Réception des données de Chirpstack sur la VM	20
Test de la communication	20
Intégration des données de l'OSUR	40
Traitement des données	265
Mise en place de NATS sur la VM	60
Développement de fonctions en Go	50
Mise en place d'un flux RSS	30
Suppression de Gluster FS	10
Mise en place de la BDD	50
Test global de l'architecture sur la VM	15
Transfert depuis la VM vers le cluster	50
Test global de l'architecture sur le cluster	15
Lien avec l'utilisateur	165
Interface utilisateur de monitoring	50
Affichage de statistiques via Jupyter et Grafana	35
Écriture du guide détaillé	80
Rapports et organisation du travail	560
Réunions	75
Rapport Planification	75
Soutenance planification	50
Soutenance mi-parcours	50
Rapport Conception	80
Page Web	80
Rapport Final	100
Soutenance finale	50
Total	1550

TABLE 2 – Tableau compilant les différentes tâches à réaliser au cours du projet



FIGURE 5 – Frise chronologique