

RAPPORT

---

# **ROULE MA POULE SANS NID-DE-POULE**

---

3 mars 2019

Bastian Bouchardon

Polytech Tours

Département informatique industrielle

# Table des matières

|         |  |    |
|---------|--|----|
| 1.1     | Introduction . . . . .   | 2  |
| 1.2     | Rappel du contexte de la réalisation . . . . .                       | 2  |
| 1.2.1   | Contexte . . . . .   | 2  |
| 1.2.2   | Objectifs . . . . .  | 2  |
| 1.3     | Livrables . . . . .  | 3  |
| 1.4     | Gestion de projet . . . . .  | 4  |
| 1.4.0.1 | Analyse de risque . . . . .  | 4  |
| 1.4.1   | Outils . . . . .   | 4  |
| 1.4.1.1 | Kanban . . . . .   | 5  |
| 1.4.1.2 | Diagramme de Gantt . . . . .   | 6  |
| 1.4.1.3 | Gestion de versions / Tests unitaires / mise en production . . . . . | 7  |
| 1.4.2   | Plannings . . . . .  | 9  |
| 1.4.3   | Déroulement du projet . . . . .                                      | 10 |
| 1.5     | Bilan financier . . . . .  | 11 |
| 1.6     | État des lieux . . . . .   | 12 |
| 1.6.1   | Livrable n° 1 . . . . .  | 13 |
| 1.6.2   | Livrable n°2 . . . . .   | 14 |
| 1.7     | Expérience personnelle . . . . .                                     | 14 |
| 1.7.1   | Difficultés rencontrées . . . . .                                    | 14 |
| 1.7.1.1 | Piézoélectrique . . . . .  | 14 |
| 1.7.1.2 | Traces GPS . . . . .   | 14 |
| 1.7.1.3 | Librairie de l'accéléromètre . . . . .                               | 14 |
| 1.7.2   | Points positifs . . . . .  | 15 |
| 1.7.3   | Points négatifs . . . . .  | 15 |
| 1.8     | Conclusion . . . . .   | 16 |

## 1.1 INTRODUCTION

Ce document donne des précisions sur le projet "Roule ma poule sans nid-de-poule", projet encadré par Emmanuel Neron et Nicolas Monmarche. Il met en évidence les liens entre chaque document constituants les livrables, les outils de gestion de projet mis en place ainsi que ma prise de recul sur le projet avec les points positifs et négatifs, les difficultés rencontrées et des conseils pour les futurs étudiants sur ce projet.

## 1.2 RAPPEL DU CONTEXTE DE LA RÉALISATION

### 1.2.1 Contexte

Aujourd'hui, de nombreux moyens de déplacement existent pour faire les trajets quotidiens comme se rendre au travail, aller faire des courses, se déplacer à des événements culturels. Il y a les transports en commun comme le bus, le tramway ou le covoiturage et des transports personnels comme la voiture, le skateboard, les appareils électriques type mono wheel, hoverboard et le vélo. Ce dernier est de plus en plus utilisé pour les trajets quotidiens car il permet de garder une bonne forme physique, il est plus rapide que la voiture en ville, peu coûteux et facile d'entretien. Les cyclistes sont sensibles à l'état de la route, en effet, les routes abîmées, écorchées peuvent au mieux endommager, au pire les blesser. En France, d'après plusieurs études, les routes se dégradent, des motards sont décédés à cause de nids de poule, des accidents ont souvent lieu impliquant des cyclistes et des défauts de la route. C'est pour ces raisons que ce projet peut être apprécié des utilisateurs de vélo.

### 1.2.2 Objectifs

Il s'agit de réaliser un système embarqué sur tout type de vélo permettant de capter les vibrations dues à la route et d'envoyer ces informations avec une position GPS sur un serveur. Cela permettra de cartographier l'état des routes et de prévenir les usagers et collectivités. Ce système sera constitué, au final, d'une carte électronique contenant l'intelligence, le capteur, le récepteur GPS, l'alimentation et la communication sans fil. Au départ, il faudra tester et choisir une technologie de capteur qui correspondra à notre utilisation. Il faudra également créer un serveur pour recevoir ces informations.

### 1.3 LIVRABLES

Pour documenter mon travail et apporter toutes les informations techniques nécessaires à l'utilisation, la maintenance, la poursuite des travaux sur ce projet, voici la liste des documents disponibles dans les livrables :

- **Cahier des charges** : Ce premier document détaille tous les besoins du client, de façon simplifiée et synthétique. J'ai participé à l'élaboration de ce document. J'ai apporté des idées et des pistes techniques. Je l'ai ensuite élaboré avec l'approbation de Nicolas Monmarche.
- **Cahier de spécifications** : Ce document est fourni avant la réalisation technique. Je décris ma prise de recul sur le projet, l'étude du cahier des charges, les détails techniques du projet global, des capteurs à l'application mobile de l'utilisateur. Je décris également les deux livrables attendus par le client. Je détail les différentes technologies de capteur de vibration qui peuvent être utilisées, les blocs fonctionnels nécessaires dans le boîtier, l'interaction entre le boîtier, l'application mobile et le serveur. Il y a également des diagrammes présentant les programmes et un exemple de design du boîtier avec des voyant, la prise de recharge ainsi que le bouton ON/OFF.
- **Cahier d'analyse** : Ce document décrit toutes les parties d'études théoriques et pratiques sur le projet. Concernant la partie électronique, différents montages d'interfaçage des capteurs sont décrits ainsi que des études logicielles concernant les librairies et la classe principale.
- **Manuel utilisateur** : Tous les détails de l'utilisation du boîtier et de l'application mobile sont présents dans ce document. Des tutoriels expliquent comment pourrait fonctionner le boîtier et l'application et comment s'en servir. Comment recharger le boîtier, comment le paramétrier, Comment voir ses parcours.
- **Documentation Conception logicielle** : Ce document présente tous les détails techniques sur le logiciel que j'ai développé pendant mon PFE. Des diagrammes de classes, des logigrammes, ainsi qu'une description détaillée du code source seront présents dans ce document. Des détails concernant la suite du projet est également disponible dans ce document. Vous trouverez également le lien GitLab pour accéder au code source.
- **Documentation Conception matérielle** : Cette documentation décrit les différents éléments électroniques choisis, toutes les caractéristiques qui les définissent et mon choix de capteur pour la surveillance des failles dans la route.

## 1.4 GESTION DE PROJET

Dans cette section, nous allons voir les outils que j'ai mis en place et pour certain expérimenté, comment j'ai échangé avec les encadrants, quelles méthodes j'ai mis en place.

Dès le début du projet, j'ai mis des outils en place que je connaissais déjà et pendant le déroulement du projet j'ai choisi d'autres outils pour les expérimenter et me faire mon opinion.

### 1.4.0.1 Analyse de risque

Après la rédaction du cahier des charges, j'ai effectué une analyse de risque me permettant d'anticiper d'éventuels problèmes durant la réalisation, avoir des solutions face à des contrebemps.

| Risques                    | Gravité /5 | Action préventives                                      | Action corrective  |
|----------------------------|------------|---|--|
| Manque de temps            | 3          | prévoir des heures en soirée pendant la semaine         | Se contenter des livrables n°1 (voir Spécifications)                               |
| Commandes en retard        | 2          | Comander le plus tôt possible                           | Concevoir en attendant les commandes   |
| Mauvaise analyse du besoin | 4          | Bien rédiger le cahier des charges, réunions régulières | Revoir les livrables avec l'encadrant  |
| Composant défaillant       | 5          | Commander les composants minimum en double              | Re-commander le composant (Radio Son si cela est durant la fermeture comptabilité) |

FIGURE 1.1 – Tableau des risques

### 1.4.1 Outils

La première tâche à réaliser était la rédaction d'un cahier des charges avec mon encadrant. Après cela, et au vu du travail à apporter j'ai décidé que le projet dans sa globalité ne pouvait pas être réalisé avec agilité car il y a une grande partie d'électronique embarqué. L'électronique est fixe, lorsqu'une carte est réalisée, nous ne pouvons pas faire marche arrière et compiler une nouvelle version, il faut refaire tout le processus de conception et réaliser une nouvelle carte, ce qui est coûteux en temps et en argent. Je devais donc penser à toutes les fonctionnalités de ce système embarqué avant de le réaliser.

Au contraire la partie logicielle, comme elle n'est pas terminée, elle pourra être réalisée avec agilité. J'ai néanmoins utilisé des outils provenant de l'agilité pour réaliser mon projet car je

trouve qu'ils sont plus efficaces pour communiquer avec les encadrants, pour visualiser le travail fait et à faire.

#### 1.4.1.1 Kanban

Pour définir toutes mes tâches provenant du cahier des charges et du cahier de spécifications, pour communiquer avec mes encadrants, tuteurs, pour lire en temps réel mon avancé et déplacer des tâches facilement, j'ai choisi d'utiliser la méthode kanban qui est souvent appliquée à l'agilité. Pour pouvoir communiquer simplement mes kanbans à tous les acteurs, j'ai décidé d'utiliser Trello.



FIGURE 1.2 – logo Trello

C'est un service disponible sous la forme d'un site internet et d'applications iPhone et Android. Il suffit de créer un compte et ensuite on peut créer des équipes, des "tableaux" ou rejoindre une équipe.

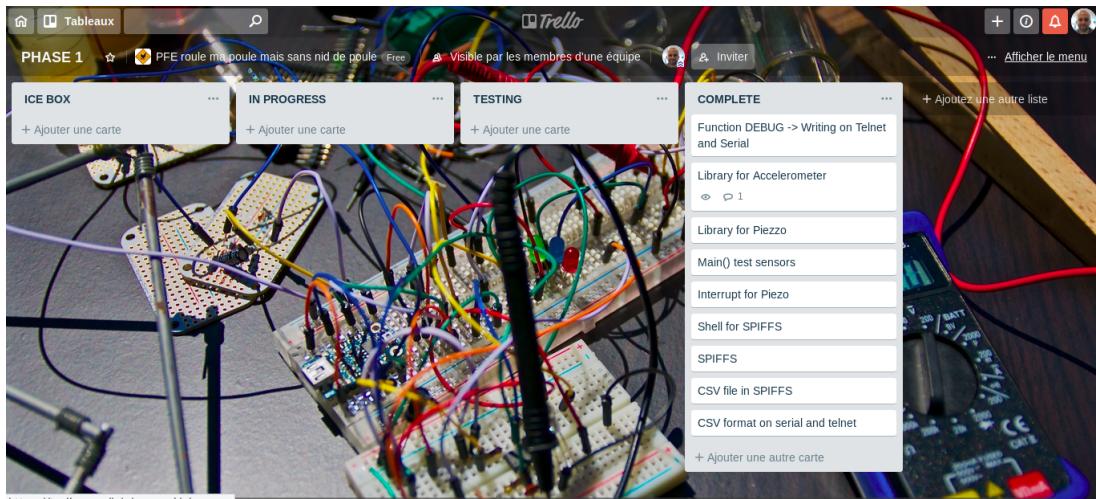
J'ai créé une équipe "Roule ma poule sans nids-de-poule", et deux tableaux :

**PFE** : Tableau principal du projet. Toutes les tâches sont présentes dans ce tableau, la rédaction des documents, la partie d'analyse, le développement logiciel, les tests, l'étude électronique sur les capteurs.

A screenshot of a Trello board titled 'PFE'. The board is organized into five columns: ICE BOX, EMERGENCY, IN PROGRESS, TESTING, and COMPLETE. 
 - ICE BOX: Contains tasks like 'Fabrication carte électronique' (due 24 janv.), 'Fabrication boîtier' (due 18 janv.), 'Création serveur' (due 23 déc. 2018), 'Programmation serveur' (due 23 janv.), and 'Conception graphique application' (due 11 janv.).
 - EMERGENCY: Contains tasks like 'Conception boîtier' (due 6 janv.), 'Programmation application' (due 28 janv.), 'Documentation maintenance' (due 1 févr.), 'Documentation utilisateur' (due 1 févr.), and 'Documentation "pour le futur"' (due 7 févr.).
 - IN PROGRESS: Contains tasks like 'Programmation Bluetooth boîtier' (due 14 déc. 2018), 'Schéma carte finale' (due 1 févr.), 'Report' (due 10 févr.), and 'Présentation soutenance' (due 13 févr.).
 - TESTING: Contains tasks like '+ Ajouter une carte' and 'PHASE 1 : Tests des capteurs' (due 19 déc. 2018).
 - COMPLETE: Contains tasks like 'Programmation GPS' (due 15 déc. 2018), 'Cahier d'analyse' (due 13 janv.), 'Programmation WiFi boîtier' (due 18 janv.), 'PHASE 1 : Conception carte prototype test capteurs' (due 23 nov. 2018), and 'Cahier des charges' (due 18 janv.). 
 The board also includes a sidebar with a camera icon and a link to a video showing a bicycle wheel.

FIGURE 1.3 – Tableau PFE (Trello)

**PHASE 1 :** Un tableau spécifique au développement logiciel me permettant de décrire plus précisément chaque tâche.



**FIGURE 1.4 – Tableau PHASE 1 (Trello)**

Les deux tableaux suivent la même logique. Il y a une ou deux colonnes avec toutes les tâches non traitées puis une colonne "in progress" où chaque tâche est actuellement traitée, en cours de traitement. Puis la tâche est déplacée dans "testing". Cette colonne indique que la tâche a été réalisée et qu'elle est maintenant testée. S'il y a des problèmes ou des erreurs alors ils sont corrigés dans cette colonne. Lorsque tout est fonctionnel dans la tâche, je la déplace dans "complete". Je la coche pour qu'elle apparaisse terminée dans le trellogramme présenté ci-dessous.

J'ai également ajouté des étiquettes à des tâches permettant d'améliorer la lisibilité des tableaux. En effet j'ai associé une couleur à un type de tâche comme des tâches concernant la documentation ou la conception logicielle.

Un autre avantage de Trello a été les mails de notifications que je recevais toutes les semaines me rappelant les tâches en cours, les retards, les avancements. Cela me permettait d'être toujours au fait de ce que j'avais à faire.

#### 1.4.1.2 Diagramme de Gantt

Pour réaliser un planning prévisionnel, définir un temps d'exécution de chaque tâche, j'ai créé un diagramme de Gantt. Pour qu'il soit en cohérence avec mes tâches de l'outil Trello, j'ai décidé d'utiliser TrelloGantt, un outil en ligne connecté à son propre Trello et permettant d'afficher nos tâches sous la forme d'un Gantt traditionnel avec une timeline.



FIGURE 1.5 – TrelloGantt logo

Cet outil est basique, ce n'est pas un PERT avec un chemin critique, des deadlines au plus tôt et au plus tard ... Il me permet simplement d'afficher mes tâches dans une timeline, de pouvoir les déplacer simplement dans le calendrier, de leur donner une durée et de pouvoir me rendre compte du travail déjà fait et de ce qu'il me reste à faire. Néanmoins, ces fonctionnalités me conviennent et je n'avais pas besoin de fonctionnalités supplémentaires pour gérer mon projet.

Voici un aperçu de mon TrelloGantt pour le tableau principal :

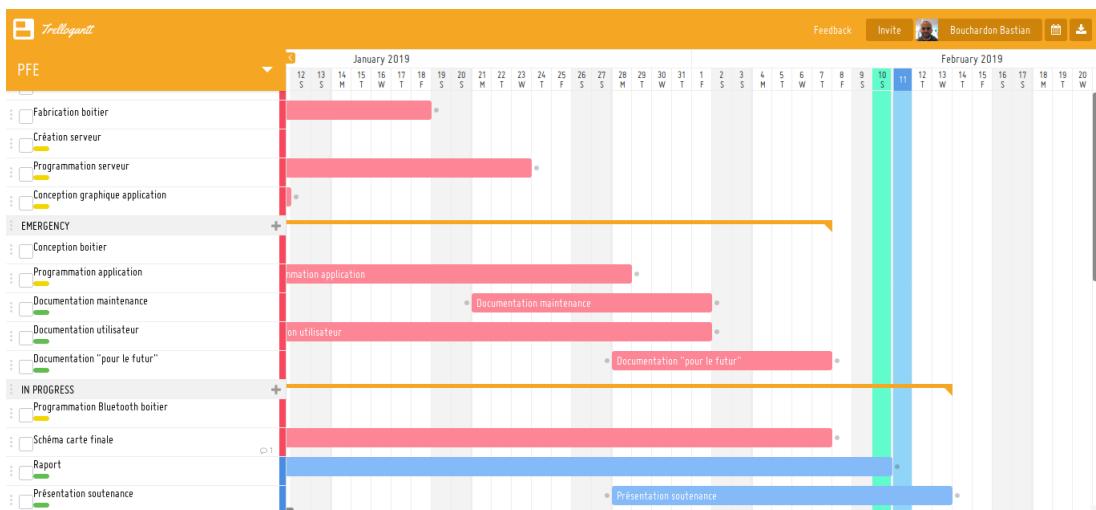


FIGURE 1.6 – TrelloGantt pour le tableau PFE

#### 1.4.1.3 Gestion de versions / Tests unitaires / mise en production

Concernant la gestion des versions de la partie logicielle du projet, j'ai décidé d'expérimenter un outil dont j'avais beaucoup entendu parler mais que je n'avais encore jamais utilisé. En effet, le code source doit pouvoir être disponible pour les prochains étudiants sur ce projet, ils devaient être hébergés sur internet. Le choix évident qui me venait à l'esprit était le service GitHub mais je voulais que le code source ne soit pas disponible pour tout le monde. Je voulais qu'il reste dans un cercle fermé, c'est pourquoi je me suis tourné vers GitLab. Ce service, comparé à GitHub est également disponible sur internet mais celui-ci, vous pouvez le télécharger et l'installer en local sur un serveur par exemple. Il se base également sur le logiciel de gestion de version Git et il est similaire à GitHub car c'est fork de ce dernier. Mais

la comparaison s'arrête ici car GitLab propose de nombreux autres services intégrés pour les utilisateurs. Il combine un outil d'agilité, un gestionnaire ticket d'incident, Git, un système de vérification de code avec l'intégration de tests unitaires, de tests de sécurité, de packaging, de mise en production et de monitoring global du projet.

|                       | Manage   | Plan  | Create  | Verify  | Secure  | Package  | Release  | Configure   | Monitor   |
|-----------------------|--|---|---|---|---|--|--|---|---|
| Separate Applications |  CA Agile Central |  Jira          |                  |  sonarqube |  JFrog Artifactory |  Electric Cloud       |  CHEF |  New Relic     |   |
| Single Application    |  Cycle Analytics  |  Issue Tracker |  Version Control |  CI        |  SAST              |  Cont. Image Building |  CD   |  Infra. Config |  Metrics |
|                       | DevOps Score   | Issue Board   | Code Review   | Unit Test   | DAST  | Container Registry   | Release Automation   | ChatOps Operations  | Infra. Monitoring APM/Tracing   |
|                       | Portfolio Mgmt   | Web IDE   | Integration-Test  | Dep. Scanning   | Binary Repository   | Pages  | Feature Flags  | Prod. Monitoring Error Tracking   | Logging   |
|                       |  |   | Acceptance-Test   | Cont. Scanning  |   | Review Apps  | App Control-Panel  |   |   |
|                       |  |   | Performance Test  | License Mgmt  | RASP  |  |  |   |   |

**FIGURE 1.7 – Outils inclus dans GitLab**

Il permet aussi de créer des projets privés, de gérer les droits utilisateurs. J'ai pu expérimenter l'exécution de mes tests unitaires automatiques à chaque commit du projet. En effet GitLab inclut le logiciel Docker permettant de tester le programme dans un environnement fermé (CI) intégration continue puis lorsque ces tests sont validés de passer à la phase livraison continue ou mise en production.

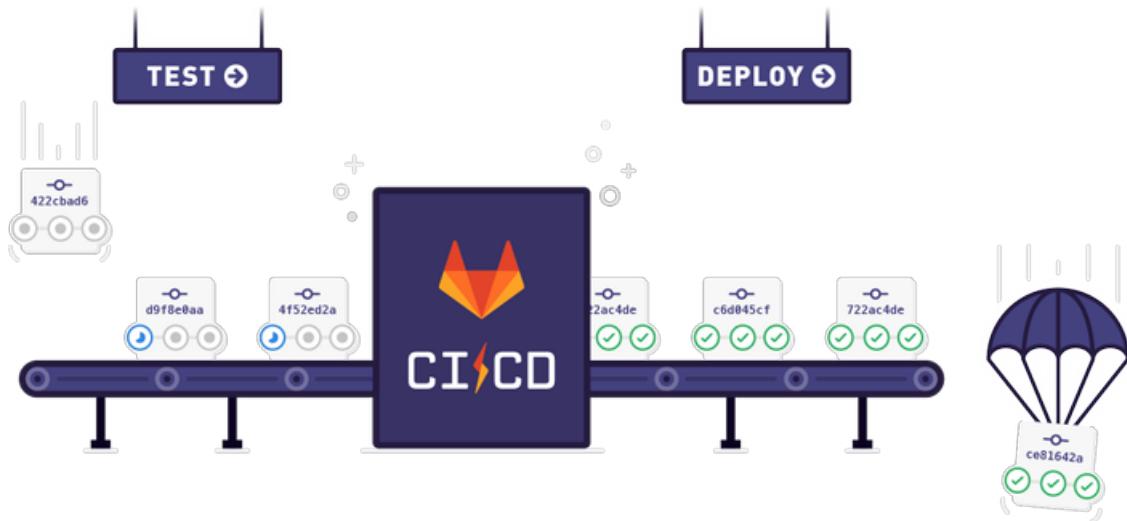


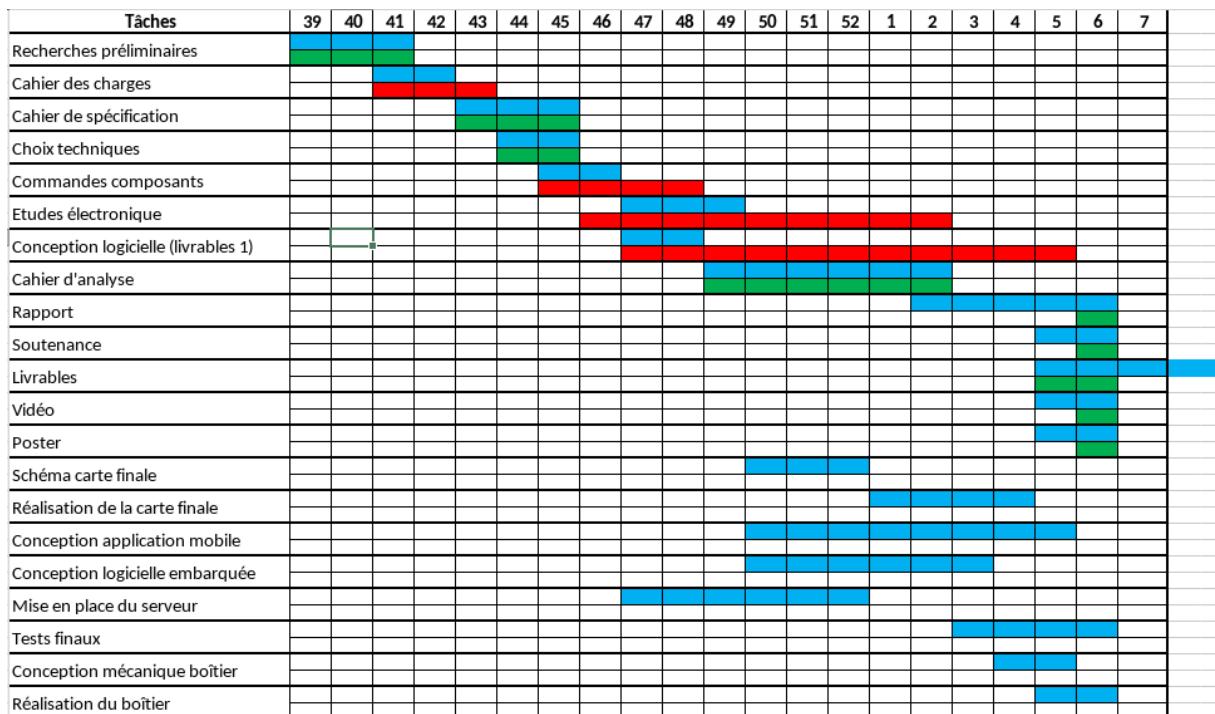
FIGURE 1.8 – Outils inclus dans GitLab

J'ai donc créé des programmes avec des fonctions de tests unitaires qui sont exécutés automatiquement à chaque commit. Si le test ne passe pas, je reçois un mail avec les résultats des tests.

#### 1.4.2 Plannings

Après l'écriture du cahier des charges et durant la rédaction du cahier de spécifications, j'ai réalisé le tableau PFE, tableau Kanban principal du projet sur Trello puis le gantt prévisionnel. Au cours du projet j'ai déplacé des tâches, fini certaines, j'ai gardé une trace de ces déplacements pour ensuite réaliser le bilan du projet d'un point de vue temporel. J'ai réalisé un gantt sur Excel en prenant les tâches principales du TrelloGantt avec en bleu la durée prévisionnelle, en vert les tâches réalisées dans les temps et en rouge les tâches qui ont du retard.

On peut voir que de nombreuses tâches ont été définies dans ce planning qui touchent l'électronique embarquée, l'informatique embarquée, la gestion réseau, de la programmation orientée objet, de la mécanique et de l'administration réseau. Pendant le projet, du fait de ne pas avoir de créneaux dans la semaine prévue pour la réalisation de notre PFE, le nombre

**FIGURE 1.9 – Planning prévisionnel / réel**

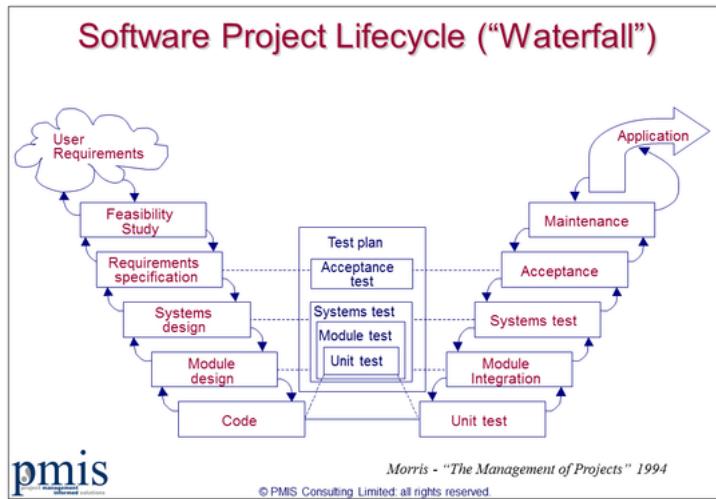
de cours, les comptes-rendus de TP, les révisions, les partiels, j'ai dû baisser mes attentes sur la réalisation du projet. En accord avec l'encadrant je me suis tenu de réaliser la phase 1 des livrables qui était la plus importante : Valider le concept et choisir un capteur qui puisse convenir à nos besoins en matière de sensibilité aux chocs.

Le fait également que nous soyons de futur ingénieur, nous concevons toujours de nouveaux systèmes, avec de la conception logicielle. C'est un travail constant de création qui rend tout planning impossible. Je l'ai d'autant remarqué que j'ai pris beaucoup de temps à tester différents types de montages électroniques et des fonctions dans mon logiciel embarqué qui seront nécessaires pour la suite du projet mais qui n'étaient pas prévus en amont.

### 1.4.3 Déroulement du projet

Comme expliqué précédemment, je n'ai pas réalisé le projet avec agilité à proprement parlé, du fait d'avoir un cahier des charges bien défini. Par contre je voyais pratiquement toutes les semaines mon encadrant pour lui rapporter mon avancement et discuter avec lui des points importants.

La partie logicielle fut plus agile avec la participation de mon second tableau : PHASE 1. J'ai pu découvrir des technologies intéressantes au projet pendant que je travaillais dessus.

**FIGURE 1.10 – Méthode Waterfall**

Cela correspondait à des User Stories que j'intégrais au fur et à mesure de la conception. J'ai pu intégrer également mes tests unitaires et les rendre automatiques avec l'aide de GitLab.

## 1.5 BILAN FINANCIER

Pour le projet j'ai doublé les capteurs pour éviter les problèmes dus à une défaillance. Voici le bilan de mes commandes pour le projet :

| Désignation produits, prestations  | Quantité | Montant unitaire H.T. | Quantité | Montant H.T. | Montant T.V.A. | Montant T.T.C. |
|------------------------------------|----------|-----------------------|----------|--------------|----------------|----------------|
| Capteur film piézo électrique      | 2        | 8.15 €                | 2        | 16.30 €      | 3.26 €         | 19.56 €        |
| Capteur type centilever 260pC/g    | 2        | 4.34 €                | 2        | 8.68 €       | 1.74 €         | 10.42 €        |
| Capteur type centilever 16pC/g     | 2        | 4.29 €                | 2        | 8.58 €       | 1.72 €         | 10.30 €        |
| Module Accéléromètre 3 axes        | 2        | 13.14 €               | 2        | 26.28 €      | 5.26 €         | 31.54 €        |
| Module ESP32 Espressif systems     | 2        | 8.73 €                | 2        | 17.46 €      | 3.49 €         | 20.95 €        |
| Module capteur vibration SW420     | 2        | 2.18 €                | 2        | 4.36 €       | 0.87 €         | 5.23 €         |
| GPS                                | 1        | 20.03 €               |          | 20.03 €      | 4.01 €         | 24.04 €        |
| Batterie 18650 Lithium-ion 2700mAh | 4        | 9.73 €                | 4        | 38.92 €      | 7.78 €         | 46.70 €        |
| TP4056 battery regulation USB      | 2        | 0.39 €                |          | 0.77 €       | 0.15 €         | 0.93 €         |
| <b>Total</b>                       | 19       |                       |          | 141.38 €     | 28.28 €        | 169.66 €       |

**FIGURE 1.11 – Bilan financier - composants électronique**

Pour la partie humaine, j'ai été seul à travailler sur ce projet, en prenant comme moyenne une journée de travail par semaine en plus de la semaine de travail en école avec des semaines plus chargées en fin d'année 2018 et début 2019, le SMIC horaire brut à 10,03€ et 21 semaines

de PFE, j'arrive à 1474 €.

| Heures travaillées / semaines | Brut / h | Nb semaines | Total      |
|-------------------------------|----------|-------------|------------|
| 7                             | 10.03 €  | 21          | 1,474.41 € |

**FIGURE 1.12 – Bilan financier - Humain**

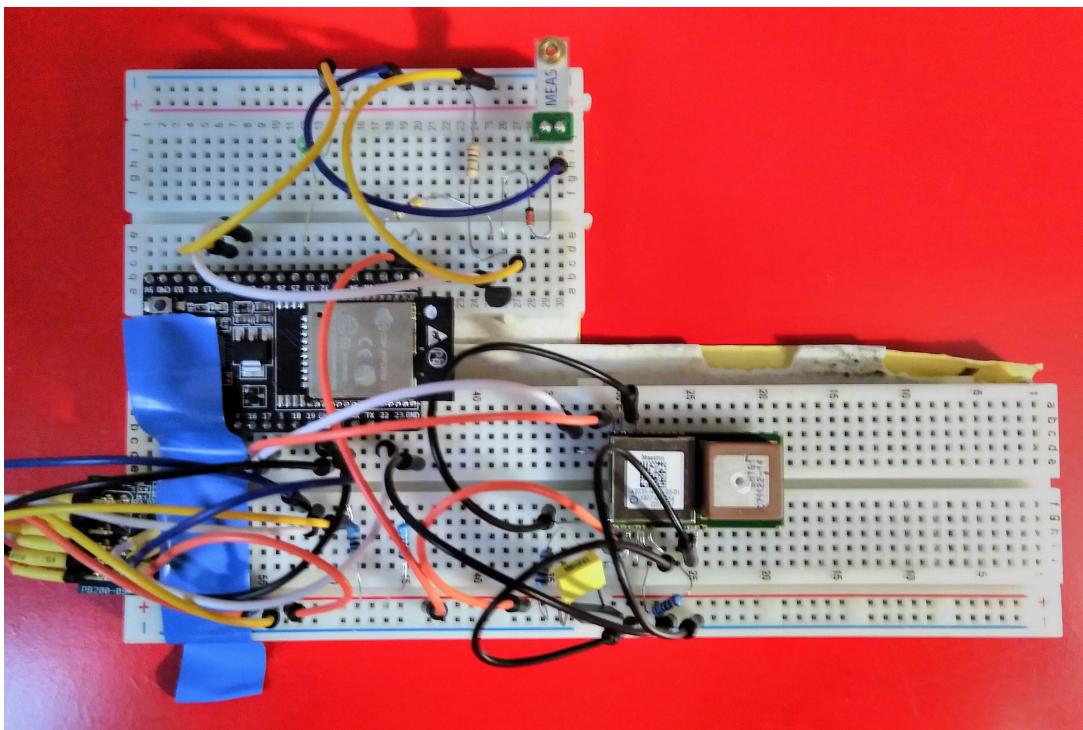
Au total ce projet aura coûté 1644,07 € pour 21 semaines de travail et un prototype pouvant enregistrer et envoyer sur port série ou telnet des fichiers CSV avec l'heure, la localisation, et les valeurs de l'accéléromètre et du piézoélectrique.

## 1.6 ÉTAT DES LIEUX

Dans ce chapitre, j'aborde tous les éléments du cahier des charges en exprimant leurs avancements. J'ai divisé en deux parties ce chapitre pour correspondre au cahier de spécification qui décrit 2 livrables, le premier concernant les tests fonctionnels de tous les éléments qui constituent le boîtier et le choix d'un capteur. Le second Livrable correspond à la réalisation du boîtier, de l'application mobile et de la création du serveur.

### 1.6.1 Livrable n° 1

En suivant le cahier de spécification et le cahier des charges, j'ai étudié les technologies de capteurs de vibrations existants, j'ai choisi 4 capteurs de vibration différents avec des formes et des technologies différentes. J'ai conçu des schémas électroniques pour tester ces capteurs. Vous pouvez retrouver ces études dans le cahier d'analyse. J'ai intégré deux capteurs, un piézoélectrique et un accéléromètre sur la platine de test de la solution. J'ai conçu une classe de test permettant de créer un réseau WiFi et un serveur telnet et/ou une liaison série. Cela permet de recevoir les informations de l'ESP32 avec ou sans fil. J'ai également utilisé une classe permettant de créer un système de fichiers sur la flash (SPIFFS) pour enregistrer le fichier CSV directement sur l'ESP32 et un shell sur le port série pour pouvoir gérer le fichier CSV. J'ai réalisé une dizaine de tests sur route et sur table pour valider le GPS, l'ESP32, les capteurs, la puissance des vibrations dû à des nids-de-poule. J'ai choisi l'accéléromètre pour plusieurs raisons évoquées dans la documentation de conception matérielle. Pour résumé, toutes les exigences du livrable n° 1 sont réalisées et validées.



**FIGURE 1.13 – Prototype Livrable n° 1**

## 1.6.2 Livrable n°2

Sur la partie 2, comme expliqué page 9, la réalisation de la carte finale, la programmation de l'application, du serveur et la conception mécanique n'ont pas été effectuées. Néanmoins j'ai réalisé un premier schéma de la carte pouvant avec tous les éléments étudiés pour le livrable n°1 que vous pouvez retrouver dans les livrables sous forme d'un projet KiCad et le détail de ce schéma dans le document de conception matérielle.

# 1.7 EXPÉRIENCE PERSONNELLE

## 1.7.1 Difficultés rencontrées

### 1.7.1.1 Piézoélectrique

Un capteur piézoélectrique génère une tension lors de sa déformation ou inversement. Cette tension, suivant la déformation peut atteindre 70V ou plus en positif comme en négatif mais avec un très faible courant. Une simple résistance de  $1\text{ M}\Omega$  peut faire chuter le signal. J'ai dû expérimenter plusieurs types de montage pour adapter ce signal à une entrée analogique ou logique d'un microcontrôleur.

### 1.7.1.2 Traces GPS

Durant les tests que j'ai effectué avec le prototype sur un vélo, j'ai rencontré des problèmes de réception des trames GPS. En effet en faisant un parcours de plusieurs kilomètres à traverser la ville, le prototype ne recevait plus de trames sur une grande partie du parcours. À l'heure où j'écris ce rapport, je n'ai pas élucidé ce problème. Je pense que c'est dû au fait d'utiliser une breadboard. Une breadboard n'est pas faite pour être utilisée dans un environnement difficile comme l'extérieur avec des vibrations. Aucun composant n'est soudé donc ils sont très sensibles aux vibrations. Il suffit d'un faux contact pendant le début d'une trame envoyé sur le port série entre le GPS et l'ESP32 pour ne pas détecter la position. Les détails de ce problème et la solution seront décrits dans le document du concepteur matériel.

### 1.7.1.3 Librairie de l'accéléromètre

Au départ, j'avais trouvé une librairie conçue par le fabricant du capteur. Après de longs essais, j'ai recréé une librairie en I<sup>2</sup>C permettant de communiquer avec le composant et cette fois-ci j'ai pu recevoir les accélérations X, Y et Z. La difficulté a été de ne pas perdre du temps à essayer de faire fonctionner cette librairie et de trouver une autre solution. Cela m'a

également permis d'avoir une librairie qui correspond exactement à mes besoins, c'est-à-dire légère car il n'y a que du code utile.

### 1.7.2 Points positifs

J'ai beaucoup aimé travailler sur ce projet, tout d'abord parce que j'ai eu l'impression que j'apportais quelque chose de nouveau, que mon encadrant avait besoin de moi pour faire ces études, je me suis senti utile. J'ai également beaucoup apprécié les échanges que j'ai eu avec Nicolas Monmarche, nous avons tous les deux apporté nos solutions, il était à l'écoute et j'ai pu expérimenter des choses sur ce projet. Techniquement, j'ai pu apprendre réellement à mettre en place des tests d'intégration continue, réaliser un projet embarqué en C++ avec mes propres librairies, les trames GPS et plus particulièrement les trames RMC. Découvrir également les capteurs piézoélectriques et les batteries Li-Ions 18650. Pouvoir travailler sur un projet qui nous concerne directement est un vrai plus et c'est pour cela que je l'ai choisi.

### 1.7.3 Points négatifs

Ma plus grosse déception fut le manque de réalisation de ma part. J'aurais voulu pouvoir réaliser beaucoup plus, je pouvais faire tout ce qu'il y avait dans le cahier des charges mais au vu de la charge de travail que nous avons eu avec les cours, c'était pour moi impossible. Je pense qu'un ingénieur ne doit pas simplement réaliser des documentations mais il doit surtout innover, apporter des solutions, réaliser du concret. Je suis déçu par la quantité de concret que j'ai réalisé. Simplement avoir une véroboard pour le prototype aurait été un plus pour moi.

## 1.8 CONCLUSION

Ce projet était divisé en deux, la première partie était principalement de l'étude pour terminer sur la partie deux avec de la réalisation. J'ai validé la première partie, commencé la seconde mais ce dont j'ai pris conscience pendant la réalisation de ce projet, c'est l'importance de la transmission. Le but n'est pas de terminer le projet mais de réaliser la quantité nécessaire de documentation, en prenant du recul, en se mettant à la place d'une personne qui découvre, pour que le projet puisse être repris où il s'est arrêté. J'ai pu développer de nouvelles compétences qui me seront utiles pour mon avenir professionnel.

Je tiens à remercier Nicolas Monmarche et Emmanuel Néron qui ont proposé ce projet et qui m'ont accompagné. Je tiens également à remercier mon tuteur académique Mathieu Delalandre pour m'avoir suivi pendant trois années et Mickaël Cadour qui me supporte depuis 4 ans.

# Table des figures

|      |   |    |
|------|---|----|
| 1.1  | Tableau des risques . . . . .                       | 4  |
| 1.2  | logo Trello . . . . .                               | 5  |
| 1.3  | Tableau PFE (Trello) . . . . .                      | 5  |
| 1.4  | Tableau PHASE 1 (Trello) . . . . .                  | 6  |
| 1.5  | TrelloGantt logo . . . . .                          | 7  |
| 1.6  | TrelloGantt pour le tableau PFE . . . . .           | 7  |
| 1.7  | Outils inclus dans GitLab . . . . .                 | 8  |
| 1.8  | Outils inclus dans GitLab . . . . .                 | 9  |
| 1.9  | Planning prévisionnel / réel . . . . .              | 10 |
| 1.10 | Méthode Waterfall . . . . .                         | 11 |
| 1.11 | Bilan financier - composants électronique . . . . . | 11 |
| 1.12 | Bilan financier - Humain . . . . .                  | 12 |
| 1.13 | Prototype Livrable n° 1 . . . . .                   | 13 |