



MASTER OF SCIENCE
IN ENGINEERING



Projet de gestion

Département : EIE
Unité d'enseignement : CM-AdvProjMgmt

Auteur Sébastien Deriaz
Professeur Vincent Bourquin
Date 14 janvier 2022

Table des matières

1	Introduction	3
2	Description du projet	3
2.1	Contexte	3
2.2	Résumé du projet	3
2.2.1	Résultat	3
3	Concept of operations	4
4	Analyse des parties prenantes	5
4.1	Organisation	5
5	Cahier des charges fonctionnel	6
5.1	Contexte	6
5.1.1	Opportunité	6
5.2	Objectif	6
5.2.1	Documentation	6
5.2.2	Critère de performance	6
5.3	Livrables	7
6	Workpackages	7
7	Planning	7
8	Estimation des coûts	9
9	Conclusion	9

1 Introduction

L'objectif de ce travail est de faire une analyse d'un projet dans le cadre du cours CM_AdvProjMgmt. Le projet choisi sera mon travail de Bachelor réalisé en 2021 à la HEIG-VD intitulé "Logiciel de contrôle d'un robot de test de modules électroniques".

Le rapport du TB¹ est disponible en annexe de celui-ci.

2 Description du projet

2.1 Contexte

Ce travail de Bachelor a été proposé par la société Betch SA. La problématique apportée est la mise en place d'un système permettant de piloter un robot de test de modules électroniques. Un précédent système spécifique avait été développé mais des modifications sur la procédure l'ont rendu obsolète.

Le but du travail a été de réaliser un système polyvalent capable de s'adapter aux contraintes futures du système.

Ce travail découle également d'une initiative personnelle visant à créer un environnement pour le contrôle d'instruments de mesure.

L'objectif est de simplifier le plus possible l'installation et l'utilisation du programme afin de faciliter l'accès à la plateforme pour les utilisateurs novices.

2.2 Résumé du projet

Le projet est, dans son entièreté, réalisé sous forme informatique, il n'y a pas de réalisation physique. Le travail est séparé en deux parties :

1. Conception d'un programme en C++ gérant la communication avec les appareils et le traitement des données
2. Création d'un package python, communiquant avec le programme C++, qui permet à l'utilisateur d'écrire des scripts de la manière la plus simple possible

Le programme en C++ possède une interface graphique pour simplifier la configuration du système. Ceci est particulièrement utile pour des utilisateurs novices.

Le but premier est le contrôle d'appareils de laboratoire (oscilloscopes, multimètres, etc...), notamment la configuration des appareils, le démarrage de mesures et la lecture de ces mesures.

L'objectif secondaire est le contrôle d'appareils "spécialisés" (comme des modules d'entrées-sorties pilotés par Modbus TCP). Ces appareils, à condition qu'ils possèdent un protocole de communication standard et ouvert, peuvent être pilotés au même titre que les équipements de laboratoire.

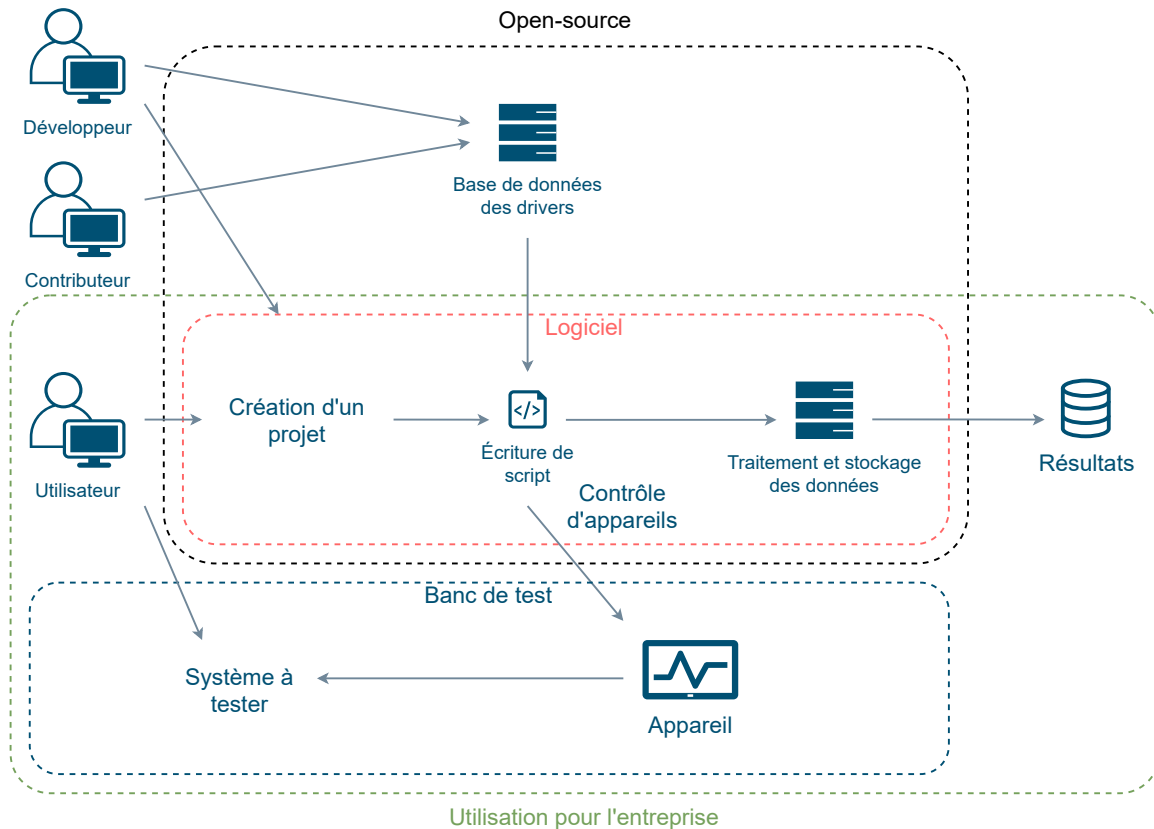
Un système de "drivers" permet de réaliser une couche d'abstraction entre le langage spécifique de chaque appareil et les commandes python. Cette partie est open-source et des contributeurs peuvent amener de nouveaux drivers.

2.2.1 Résultat

Le proof of concept s'est montré fructueux et le projet final a pu être mené à bien. Il a d'ailleurs été mis en œuvre pour la production et la caractérisation de modules électroniques dans le semestre suivant le travail de Bachelor (semestre d'automne).

1. Travail de Bachelor

3 Concept of operations



Le ConOps² mets en évidence les acteurs principaux du projet :

1. Développeur
2. Contributeur
3. Utilisateur

L'utilisateur va se servir du logiciel afin de réaliser des tests (sur un matériel et un système qui lui appartiennent et sur lesquels le développeur et le contributeur n'ont pas d'impact). Au travers de ses tests et mesures, l'utilisateur va créer des données dont le traitement et le stockage sont sous sa responsabilité.

Le développeur se charge de fournir un programme fonctionnel, qui répond au mieux aux besoins du client. Le contributeur fournit un service bénévole permettant d'étendre les capacités du système (compatibilité avec plus d'appareils par exemple).

Le workflow typique de l'utilisateur consiste en :

1. Création d'un projet
2. Écriture de code (script) permettant d'effectuer les tests et mesures dont il a besoin
3. Traitement et stockage des données
4. Publication et utilisation des données au sein de l'entreprise.

4 Analyse des parties prenantes

Les parties prenantes sont les suivantes :

1. Professeur encadrant le projet
2. Entreprise partenaire
3. Étudiant / développeur
4. Utilisateur du projet
5. Contributeur

Professeur : L'objectif du professeur est de guider l'étudiant pour l'amener à faire les bons choix et s'assurer que le projet soit sur la bonne voie

Entreprise partenaire : L'entreprise cherche à obtenir un produit fonctionnel qui rempli le cahier des charges. L'orientation open-source du projet ne profite pas à l'entreprise mais plutôt aux utilisateurs privés et aux écoles.

Étudiant / développeur : L'étudiant doit mener à bien le projet et garantir qu'il rempli le cahier des charges. En parallèle il doit s'assurer que les objectifs de l'entreprise, même si ils ne sont pas explicitement décrits dans le cahier des charges, soient rempli dans le cadre du projet (par exemple modifier le système pour prendre en compte un point évident qui n'aurait pas été mentionné dans le cahier des charges)

Utilisateur du projet : L'utilisateur a un rôle critique et va fournir un feedback utile pour déceler les défauts de conception du système autant sur le plan ergonomie que fiabilité.

Contributeur : Le/Les contributeur(s) agissent dans un second temps, dans le cas où le système est fonctionnel. Ils améliorent le système par le biais de l'open-source en fournissant des morceaux de codes permettant d'étendre les capacités du système.

4.1 Organisation

La gestion de projet sera, dans un premier temps, consacrée uniquement aux 4 premières parties prenantes du projet, afin de mener à bien la création d'un prototype.

5 Cahier des charges fonctionnel

5.1 Contexte

Lorsqu'un banc de test est réalisé (par exemple pour mesurer des systèmes et/ou effectuer des mesures), des instruments de mesures standards sont utilisés (multimètres, oscilloscopes, générateurs de signaux, etc...). La quasi-totalité de ses instruments sont conçus avec des options permettant de les contrôler depuis un ordinateur et/ou d'autres appareils. Les solutions proposées par les fabricants sont souvent limitées ou très complexes et coûteuses. Des alternatives comme LabView permettent également le contrôle de ces appareils mais le coût est important et les capacités limitées. Ce logiciel est très adapté à des fins didactiques pour des utilisateurs novices (notamment à la programmation).

5.1.1 Opportunité

Les systèmes de communication utilisés pour piloter les instruments de mesure ont été standardisés au fur et à mesure des années. Les instructions qui peuvent être envoyées aux instruments afin de les piloter sont également disponibles librement depuis le site internet du fabricant. En parallèle, des langages de programmation haut-niveau comme Python et Matlab permettent de réaliser des traitements complexes de données complexes de manière simple et efficace. Un système capable de relier ces deux milieux pourrait servir aux entreprises pour la création de bancs de tests, au privé pour effectuer de l'automatisation de mesure (et de tirer au mieux parti des capacités de leurs instruments) ainsi qu'aux écoles pour simplifier la prise de mesures (lors de laboratoires par exemples).

5.2 Objectif

Le système doit être capable, au moyen de code³, de configurer un appareil, de lancer des mesures et de récupérer des valeurs.

Les valeurs récupérées doivent pouvoir être traitées facilement puis, si nécessaire, renvoyées au même ou à un autre appareil.

Le système doit également être capable de transférer des données de sources externes vers les appareils et inversement.

5.2.1 Documentation

Cette façon de procéder, qui s'approche de l'"infrastructure as code"⁴, permet de décrire un fonctionnement physique par un code informatique, compact et commentable. Ceci permet d'effectuer le test et sa documentation d'un seul coup. Cette approche permet également de recommencer une action à un moment ultérieur sans risquer d'oublier la séquence et/ou de se baser sur une documentation incomplète (le code est répétable).

5.2.2 Critère de performance

Le premier critère permettant de valider la performance du système est sa capacité à contrôler les appareils. A l'exception d'un simple test, il est nécessaire, de part la nature complexe de la communication entre ordinateur et appareils, d'effectuer des tests rigoureux sur une longue durée afin de valider le bon fonctionnement du système (pas de pertes de données, de commandes ratées, etc...).

Le deuxième critère est la facilité de mise en œuvre de nouvelles fonctions et/ou la capacité d'évolution du système. Ce critère devra être évalué lorsque le système est mis en place dans des cas réels et que de nouvelles fonctionnalités sont ajoutées. La réussite est prononcée lorsqu'il est possible de rajouter des fonctionnalités sans altérer des précédents tests (que les codes soient réutilisables malgré l'ajout de nouvelles fonctionnalités).

3. Python, Matlab ou autre

4. https://en.wikipedia.org/wiki/Infrastructure_as_code

5.3 Livrables

Les livrables sont sous la forme d'un programme (ou un ensemble de programmes) qui peuvent être installés sur un poste de travail. Une documentation est également livrée afin de permettre à l'utilisateur de débiter.

6 Workpackages

Les workpackages pour le projet sont les suivants

WP1 : Gestion de projet

WP2 : Analyse des protocoles / Analyse des appareils

WP3 : Conception du prototype : Implémentation et réalisation

WP4 : Tests (par le développeur et par un agent externe)

WP5 : Gestion des contributeurs (open-source)

Gestion de projet : Affectation du travail, création du planning et coordination des parties prenantes

Analyse des protocoles et appareils : Recherche de l'état de l'art en terme de communication et de contrôle dans un premier temps. Dans un deuxième temps il s'agit de déterminer si et comment les protocoles de communication peuvent être implémentés dans le programme

Conception de prototype : Il s'agit de développer la base du système sur laquelle s'ajoute la couche de gestion de la communication et des protocoles, développée dans la phase d'analyse.

Tests : Les tests sont séparés en deux parties : tests par le développeur pour vérifier que l'implémentation des protocoles est fonctionnelle et tests par l'utilisateur qui permettront de valider l'ergonomie et la fiabilité du système.

Gestion des contributeurs : Cette partie se déroule après que le projet ait été livré. Il s'agit de gérer le travail effectué bénévolement par chaque contributeur, notamment quelles parties sont publiques et modifiables et quelles parties du programme sont fixes. Il convient également de s'assurer que le travail effectué par chacun respecte une logique de développement harmonieuse

7 Planning

Le projet est séparé en deux phases :

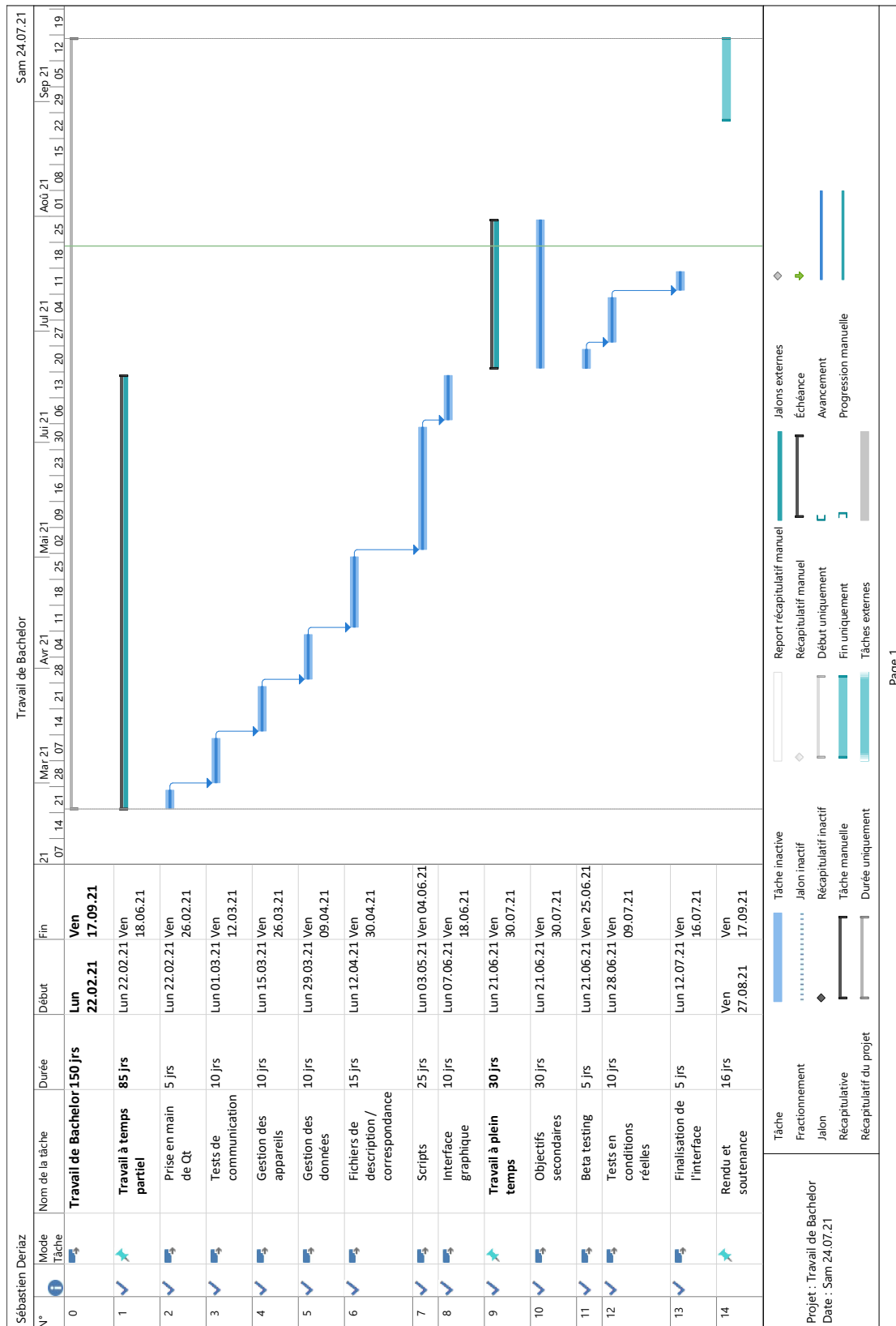
1. Temps partiel (réalisé en parallèle avec les autres cours)
2. Temps plein (après la fin des cours)

Ces deux phases seront représentées par deux barres bleues sur le Gantt (voir page suivante).

Les dates clés sont

1. 22.02.2021 : Début du projet
2. 21.06.2021 : Début du travail à plein temps
3. 27.08.2021 : Rendu du projet
4. 17.09.2021 : Soutenance de projet

Lors du travail à plein temps, il a été prévu de travailler en parallèle sur les objectifs secondaires ainsi que sur le beta-testing. Si nécessaire, le temps alloué aux objectifs secondaires peut être utilisé pour absorber un éventuel retard sur le projet



8 Estimation des coûts

Comme le projet a été réalisé dans une école, l'estimation des coûts n'as pas été réalisée au moment du développement. Le suivant calcul permet d'estimer le coût d'un tel projet si il devait être réalisé dans une entreprise. Ce calcul permet également de donner une idée du potentiel coût pour la reprise du projet.

Description	Calcul	Total
Développeur polyvalent	$90\,000 \text{ CHF} \times 0.25^5$	22 500 CHF
Utilisateur / Testeur	$90\,000 \text{ CHF} \times 0.01$	9000 CHF
Coût de l'équipement (matériel spécifique)	forfait pour matériel	3000 CHF
Coût des licences	logiciels open-source	0 CHF
Coût des Locaux (locaux estimés à 300 CHF par mois)	$300 \text{ CHF} \times 12 \times 0.25$	900 CHF
		35 400 CHF

TABLE 1 – Estimation des coûts

9 Conclusion

L'exercice de réaliser la gestion de projet à la suite de sa réalisation est un exercice difficile, particulièrement lorsque le projet est réalisé par une seule personne.

L'estimation des coûts a été réalisée à titre d'exercice afin d'avoir une idée du coût potentiel du projet. Comme le travail de Bachelor a été réalisé dans une école et que tout le matériel nécessaire était présent sur place, les coûts n'as pas été pris en compte lors du développement. De plus, comme il s'agit d'un projet informatique il n'y a pas eut de commande de matériel nécessaire.

Il en est de même pour la division en workpackages. Elle n'as pas été effectuée au début du projet car c'est un projet personnel.

5. Un quart d'année, estimé à partir du temps équivalent que représente un travail de Bachelor