|  |
| --- |
| UNIVERSITÉ TOULOUSE III PAUL SABATIER |
| Dossier de programmation |
| A1110 : Informatisation du processus d’organisation d’un triathlon (IronTask) |
| **BRESOLIN Maxime**  **HOUTMANN Hadrien**  **POUSSARD Sébastien**  **ROMERO Diego**  **TROTTET Julien** |
| 8 juin 2018 |
| |  | | --- | |  | |
| **Tuteur/Client : M Thierry Beltran** |
| *Résumé : Ce document décrit les méthodes de travail utilisées dans le cadre de notre projet. Ce document est à destination du client.* |

**Identification :**

Type de document : Document de conception

Emetteur : STEELCORP

Version : 1.0

**Auteurs :**

|  |  |  |
| --- | --- | --- |
| **Nom et prénom** | **Fonction** | **Organisme** |
| POUSSARD Sébastien | Responsable organisationnel | STEELCORP |
| ROMERO Diego | Responsable technique | STEELCORP |
| BRESSOLIN Maxime | Responsable communication | STEELCORP |
| HOUTMANN Hadrien | Responsable technique | STEELCORP |

**Vérification :**

|  |  |  |
| --- | --- | --- |
| **Nom et prénom** | **Fonction** | **Organisme** |
| BELTRAN Thierry | Tuteur | Equipe enseignante |

**Approbation :**

|  |  |  |
| --- | --- | --- |
| **Nom et prénom** | **Fonction** | **Organisme** |
| TROTTET Julien | Chef de projet | STEELCORP |

**Listes des version et révisions :**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Objet** |
| 1.0 | 21 mai 2018 | Version initiale |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table des matières**

[1. Introduction 5](#_Toc514677959)

[2. Redéfinition de l’analyse fonctionnelle 5](#_Toc514677960)

[2.1 Recherche des fonctions de services 5](#_Toc514677961)

[2.2 Caractérisation des fonctions 6](#_Toc514677962)

[3. Processus organisationnel 7](#_Toc514677963)

[3.1 Méthode de travail 7](#_Toc514677964)

[3.1.1 Mise en place de la méthode agile 7](#_Toc514677965)

[3.1.2 Présentation de l’équipe Agile 7](#_Toc514677966)

[3.1.3 Utilisation de la technologie GIT 7](#_Toc514677967)

[3.2 Planification du projet 8](#_Toc514677968)

[3.2.1 Planification prévisionnelle 8](#_Toc514677969)

[3.2.2 Planification réelle 8](#_Toc514677970)

[4. Conception du logiciel 8](#_Toc514677971)

[4.1 Conception de la base de données 8](#_Toc514677972)

[4.2 Cadre de travail 9](#_Toc514677973)

[4.2.1 Choix des technologies 9](#_Toc514677974)

[4.2.2 Modèle MVT (Modèle Vue Template) 9](#_Toc514677975)

[4.3 Détails du récit utilisateur, « Ajouter un triathlon » 9](#_Toc514677976)

[4.3.1 Les Liens (URL) 9](#_Toc514677977)

[4.3.2 Le Modèle 9](#_Toc514677978)

[4.3.3 La Vue 9](#_Toc514677979)

[4.3.4 Le Template 9](#_Toc514677980)

[5. Bilan 9](#_Toc514677981)

[6. Conclusion 9](#_Toc514677982)

# Introduction

L’équipe projet a été restructurée et est maintenant composée de 5 étudiants à l’IUT informatique (Année Spéciale). Il a alors été décidé de fusionner les cahiers des charges des différents groupes afin de répondre au mieux à la demande.

Les premiers rendez-vous avec le client, nous ont permis d’affiner les fonctions logicielles importantes et d’épurer celles qui n’étaient pas voulues par Monsieur BELTRAN.

Le groupe est chargé du développement du logiciel IRONTASK, qui permettra de faciliter l’organisation de triathlon et d’organiser les tâches inhérentes aux l’événements.

Ce document de conception décrit la structure du logiciel et ses fonctionnalités, l’architecture (base de données) et la programmation. Nous détaillons également les méthodes de travail utilisées qui ont un impact direct avec le client.

# Rappel de l’analyse fonctionnelle

Nous avons redéfini notre équipe projet. Nous sommes maintenant 5 étudiants, il a été donc décider de refaire l’analyse fonctionnelle du projet afin que nous ayons tous les mêmes fonctions contraintes.

A terme, nous avons traduit cette analyse fonctionnelle en récit utilisateur. Ces récits utilisateurs sont utilisés pour la méthode agile que nous détaillons plus loin.

## Recherche des fonctions de services

Nous avons recherché les fonctions du logiciel de gestion de triathlon. Le diagramme “pieuvre” met en évidence les relations entre les différents éléments du milieu environnant et le produit.

**FP1**

FC6

FC7,8

FC9

FC10

FC1,2,3,4,5

Figure 1 – Diagramme Pieuvre IRONTASK

Voici la liste de la fonction principale et des fonctions contraintes en référence au diagramme pieuvre ci-dessus :

|  |  |
| --- | --- |
| Id. | Désignation |
| FP1 | **Permettre à l’organisateur de faciliter l’organisation d’un triathlon** |
| FC1 | Permettre à l’organisateur d’administrer les tâches d’un triathlon (ajout, suppression, …) |
| FC2 | Le produit doit être capable d’effectuer une recherche de tâches. |
| FC3 | Doit avoir un outil d’affichage des tâches dans le temps. |
| FC4 | Doit permettre de contacter les responsables de tâches. |
| FC5 | Alerter l’organisateur des tâches en retard. |
| FC6 | Répertorier et archiver les participants sur le long terme. |
| FC7 | Doit gérer les stocks des ressources matériels (Désignation, Quantité, Lieux, …) |
| FC8 | Doit stocker les devis et autres documents. |
| FC9 | S’adapter aux différents supports (PC, Smartphone, …). |
| FC10 | Respecter les normes de la FFTRI et CNIL. |

Tableau 1 - Fonctions de services

## Caractérisation des fonctions

A présent, il est nécessaire de définir les fonctions de service (Tableau 2) avec un plus haut niveau de précision. Le tableau ci-dessous répertorie ces fonctions ainsi que le/les critères caractérisant ces fonctions en appliquant également un niveau et flexibilité pour les différents critères.

|  |  |  |  |
| --- | --- | --- | --- |
| Identifiant | Critères | Niveau | Flexibilité |
| FP1 | Gérer toutes les tâches  Planification | N tâches de bases  Calendrier année N | Ajout par FC1  -2 ans + 10 ans |
| FC1 | Ajouter/ Modifier /Supprimer une tâche | 1000 tâches | +- 100 |
| FC2 | Filtres de recherche | 5 filtres max |  |
| FC3 | Calendrier | -2ans/+5ans |  |
| FC4 | Contact par mail  Contact via messagerie interne |  |  |
| FC5 | Alerte automatique | X : paramètre tâche |  |
| FC6 | Statistique, nombres de participants  Durée d’archivage | 2 ans |  |
| FC7 | Gestion de stock (type, localité, quantité) | 2000 objets | +- 200 |
| FC8 | Stocker les devis | 512 Mo | <2048 Mo |
| FC9 | S’adapter à tous les écrans | Responsive design |  |
| FC10 | Normes FFTRI et CNIL | Normes FFTRI et CNIL | Normes FFTRI et CNIL |

Tableau 2 - FONCTIONS DE SERVICES, CRITERES, NIVEAU ET FLEXIBILITE

# Processus organisationnel

## Méthode de travail

### Mise en place de la méthode agile

« Les méthodes agiles sont des groupes de pratiques de pilotage et de réalisation de projets. Elles ont pour origine le manifeste Agile, rédigé en 2001, qui consacre le terme « agile » pour référencer de multiples méthodes existantes.

Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles, impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles reposent sur un cycle de développement itératif, incrémental et adaptatif et doivent respecter quatre valeurs fondamentales déclinées en douze principes desquels découlent une base de pratiques, soit communes, soit complémentaires. » [[1]](#footnote-1)

Dans ce cadre nous avons utilisé le Framework SCRUM, ce Framework vise à régulièrement produire des livrable au client afin de pouvoir avoir un retour régulier. Ainsi le travail sur le projet à été découpé en différents sprints sanctionné par un livrable.

Pour l’organisation des Sprints nous avons utilisé Trello [[2]](#footnote-2). Chaque sprint étant d’une durée de 2 semaines, sanctionné par un test avec le client.

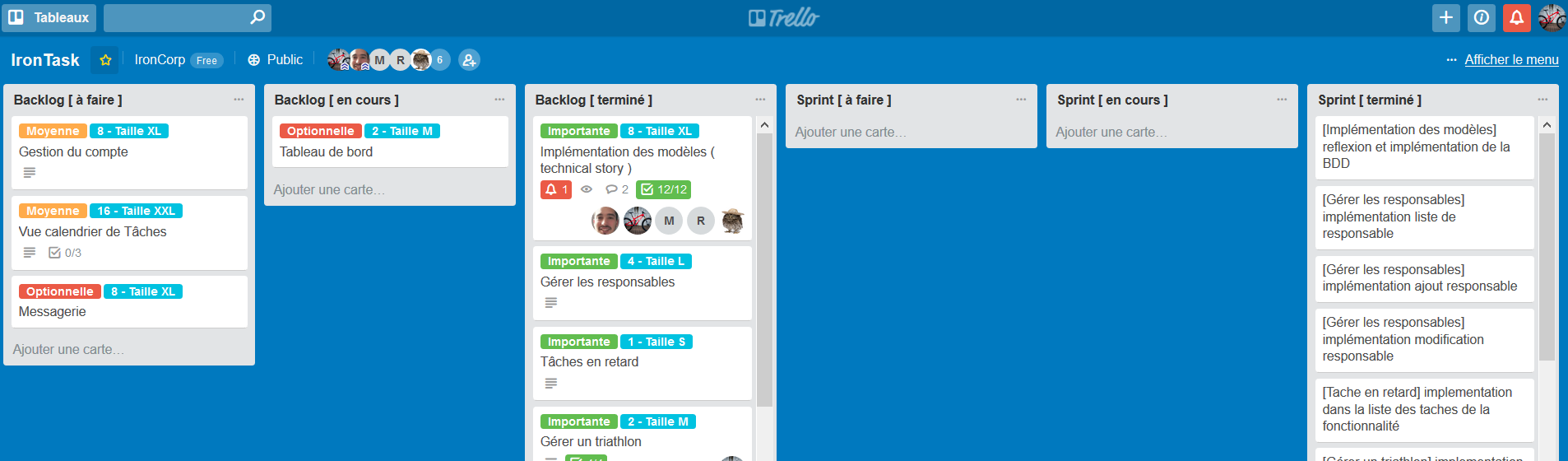


Figure 2 - Organisation sous Trello

JULIEN

Pour définir le backlog produit, redéfinition des FC en user story et technical story…

Avantage, autonomie et régularité et itératif, fonction marche de A à Z, bien pour le client

### Présentation de l’équipe Agile

Scrum-master : Julien Trottet

Product-owner : Hadrien Houtmann

Client : Thierry Beltran

Développeurs : Julien Trottet, Diego Romero, Maxime Bresolin, Hadrien Houtmann, Sébastien POUSSARD

JULIEN

Equipe autonome et homogène (merci openclassroom) avec un leader technique Hadrien Houtmann

### Utilisation de la technologie GIT

Afin de développer à plusieurs sur le même projet il est nécessaire d’adopter une méthode d’organisation. Nous avons développé régulièrement en groupe le week-end car nous avons ressenti que nous progressions plus ensemble. Cependant en semaine l’équipe développait séparément. Dans ce cadre, il a été nécessaire d’adopter l’utilisation de GIT[[3]](#footnote-3) («git est un logiciel de gestion de versions décentralisé »[[4]](#footnote-4) ).

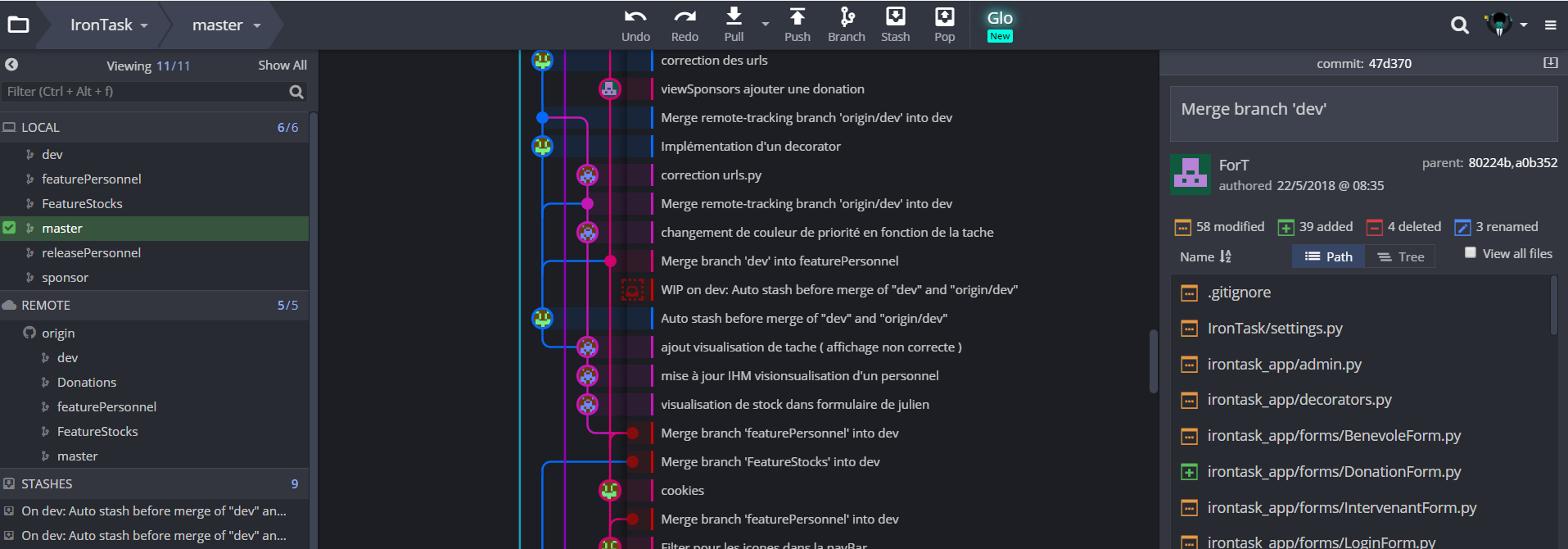


Figure 3 - Organisation code sous Gitkraken

3 blocks : Stock, Tache, Personnel (release et feature)

|  |  |  |
| --- | --- | --- |
| Type de branche | Qui valide ? | Fréquence |
| Dev | Tous | Très souvent (journalier) |
| Feature… | Tous | Souvent (hebdomadaire) |
| Release… | Scrum master et Product Owner | Rarement (un sprint, bimensuel) |
| Master | Scrum master et Product Owner | Très rarement (mensuel) |

Tableau 3 - Procédure d'utilisation des branches

### Communication

JULIEN + HAD

Slack pour le dév, réunion hebdomadaire physique sur nouvelle notion en hyperprogramming, mail rdv client.

## Planification du projet

### Planification prévisionnelle

SEB + MAXIME

### Planification réelle

SEB + MAXIME

# Conception du logiciel

## Conception de la base de données

Figure 4 : MCD de la base de donnée

Un Triathlon est caractérisé par une Type de triathlon afin de pouvoir réutiliser les différents type de triathlon (Ironman, long, short …).

Pour chaque triathlon peut intervenir des intervenants (médecins, animateurs …) étant donné que ces interventions peuvent être payantes nous avons choisis de stocker les Devis (sous forme binaire) et les prix des devis afin de l’afficher pour l’utilisateur.

A chaque triathlon sont associés également des sponsors qui vont faire des donations.

Chaque triathlon appartient également à une catégorie qui qui donne l’intervalle des âges ainsi que les sexes autorisés à participer.

Les taches seront associées à un triathlon et auront à chaque fois un responsable (qui est un bénévole) mais aussi une liste de bénévoles qui y participent.

Enfin nous enregistrons également le matériel disponible par l’association, ce matériel peut être dispatché dans différents triathlons via l’allocation notons qu’un bénévole gère ce matériel alloué, il en est le responsable.

La base de données utilise le format MySQL.

HAD

## Cadre de travail

### Choix des technologies

HADRIEN + DIEGO

### Modèle MVT (Modèle Vue Template)

JULIEN + HADRIEN

## Détails du récit utilisateur, « Ajouter un triathlon »

### Les Liens (URL)

MAXIME

### Le Modèle

Django est basé sur un ORM (Mapping objet-relationnel)[[5]](#footnote-5), qui correspond à une couche d’abstraction entre l’objet produit par l’ORM et la base de donnée relationnel.

L’ORM permet ainsi de manipuler la base de donnée à travers des objets, pour ce faire il est nécessaire de définir dans un fichier models.py les différentes tables de la base de données qui seront accessible à travers des objets dans les autres parties de Django.

Un modèle est donc une manière de décrire une table, pour ce faire il est nécessaire d’importer dans models.py différentes classes :

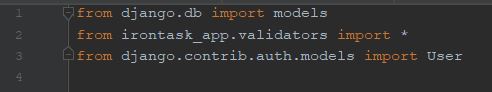


Figure 5 : import de models.py

* Models de django.db est nécessaire pour pouvoir utiliser l’ORM et définir la base de donnée dans ce fichier.
* VALIDATORS ???
* User de django.contrib.auth.models permet de gérer de manière automatique des compte utilisateur capable de se connecter à l’application. En l’état actuel seul le compte administrateur est capable de se connecter à l’application

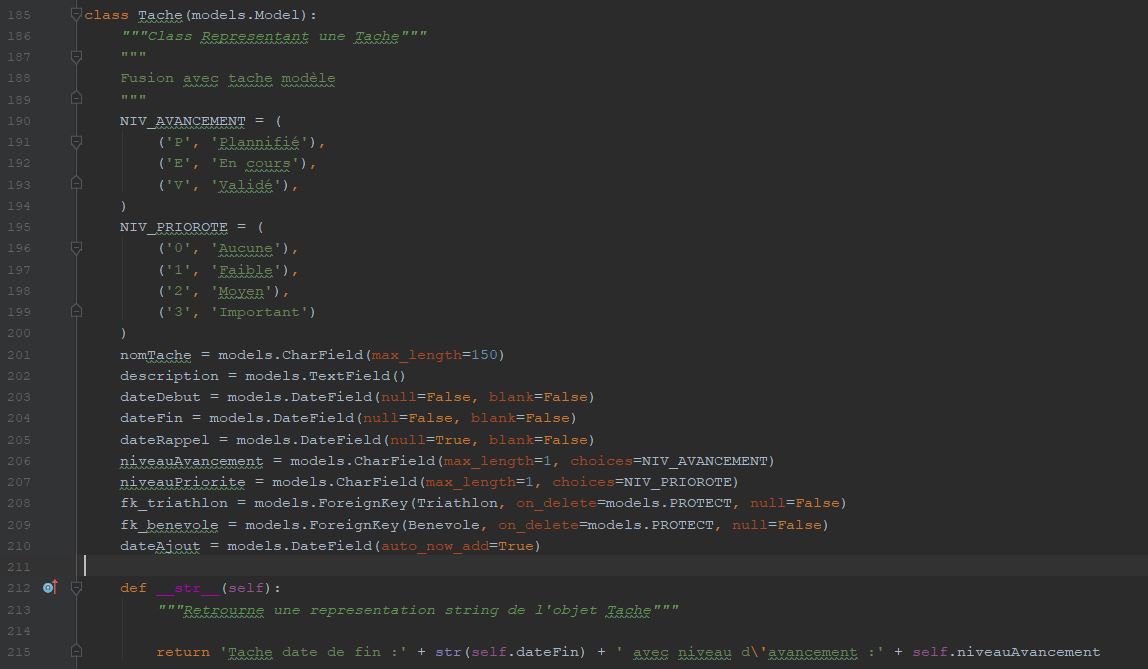
Nous pouvons constater que la table de Tache est définit comme une classe.

Figure 6 : DEFINITION DE LA TABLE TACHE

Le niveau d’avancement et le niveau de priorité sont définit comme des type de données pour faciliter leur utilisation ( une tache « planifié » est représenté en base de donnée par « P » ) cela permet d’une part d’optimiser la base de donnée mais aussi de restreindre les état possible de la variable.

Chaque colonne de la base de donnée est définit comme une variable via une fonction de type models.xxxxx :

* CharField pour les Char.
* TextField pour un Texte.
* DateField pour une date.
* ForeignKey pour une clé étrangère.
* D’autres types de fonctions ont été utilisés qui ne seront pas développés ici.

Remarque : la création de clé primaire est automatique par défaut ( elle peut toutefois si besoin est être redéfinis.

Différents paramètres peuvent être donnés pour les fonctions :

* max\_length définit la taille.
* null autorise les valeur null ou non.
* blank autorise un contenu vide ou non.
* choices permet de restreindre les valeur aux types précédemment définit (NIV\_AVANCEMENT, NIV\_PRIORITE).
* Pour les clé étrangère il est nécessaire de préciser en paramètre la table qui est indexé.
* On\_delete permet de paramétrer le comportement de la base si la clé référence par la clé étrangère vient à être supprimée, models.protect permet de protéger de la suppression en cascade.
* Auto\_now\_add permet pour une date un ajout automatique du contenue de la date à aujourd’hui.

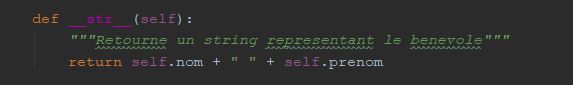
Def \_\_str\_\_ (self) est une fonction qui permet de renvoyer un String pour l’objet créer représentant la table, c’est l’équivalent du toString pour java. Cette fonction est par défaut implémenté dans Django est peut être surchargé au besoin, exemple figure suivante :

Figure 7 : surcharge de \_\_str\_\_ pour la table benevoles

Ainsi l’objet « bénévole » qui permet de travailler sur la table bénévole de la base de données pourra à travers l’appel de la méthode \_\_str\_\_ remonté comme un String définit par le nom et le prénom du bénévole.

### La Vue

JULIEN

### Le Template

DIEGO

# Bilan

# Conclusion

# 

1. https://fr.wikipedia.org/wiki/M%C3%A9thode\_agile [↑](#footnote-ref-1)
2. https://trello.com/b/op3sFIJV [↑](#footnote-ref-2)
3. https://github.com/SteelCorp [↑](#footnote-ref-3)
4. https://fr.wikipedia.org/wiki/Git [↑](#footnote-ref-4)
5. https://fr.wikipedia.org/wiki/Mapping\_objet-relationnel [↑](#footnote-ref-5)