



TANOOKI

Le coaching pour tous



CAHIER DES CLAUSES TECHNIQUES PARTICULIÈRES Client – Formation Supérieure

N° - TNK/DEV/CCTP-20200615
Date – Juin 2020

VOTRE CONTACT

Cédric Obejero

+33 (0)6 78 68 71 29

cedric.obejero@tanooki.fr

www.tanooki.fr

Clause de confidentialité

Toute information contenue dans le présent document strictement confidentiel est fournie dans le seul but de répondre aux enjeux de formation, et ne peut être utilisée à d'autre fin.

Tout destinataire s'engage à ne diffuser ni faire connaître tout ou partie des informations contenues dans le présent document à quelque tierce partie que ce soit, sans l'autorisation écrite préalable de la société TANOOKI.

Copyright © 2020

Tous droits réservés

SOMMAIRE

1. Présentation générale	3
1.1 Présentation du client	3
1.2 Présentation du projet	3
1.3 Fonctionnement de la partie opérative ROBIOT	4
1.4 Fonctionnement de la partie commande	5
1.5 Documents applicables et de référence	5
2. Expression du besoin	6
2.1 Positionnement du simulateur	6
2.2 Objectifs et besoins logiciels	6
2.3 Exigences fonctionnelles	6
2.3.1 Partie commande	6
2.3.2 Partie opérative	7
2.3.3 Partie mesure	7
2.4 Exigences non-fonctionnelles	7
3. Cadre de prestation	8
3.1 Cadre d'exécution des travaux	8
3.2 Qualité logicielle	8
3.3 Livrables de prestation	8
3.4 La gestion de projet / Planification	9
4. Annexes	10
4.1 Exemple de données de simulation – CARTOGRAPHIE TERRAIN	10
4.2 Exemple de données de simulation – LISTE DE CONTROLE	10

1. Présentation générale

Le présent document fournit les éléments de cadrage d'un projet de développement logiciel effectué dans le cadre d'une formation de qualité logicielle.

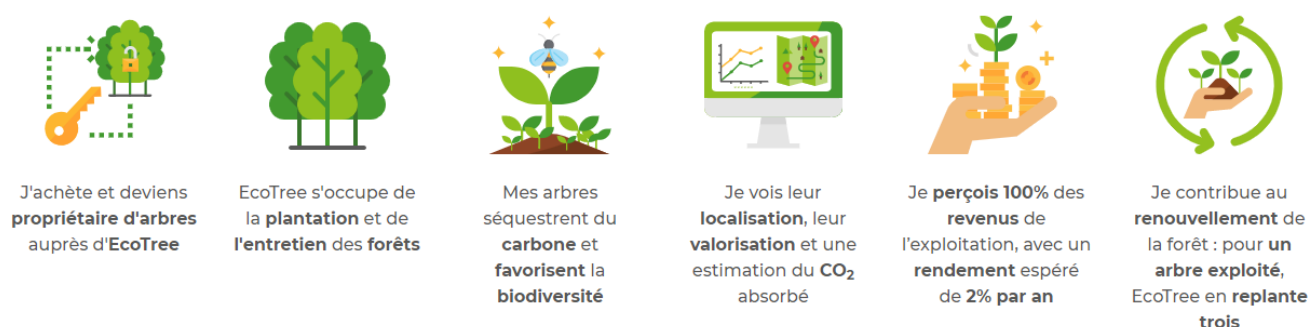
1.1 Présentation du client

Créée par cinq amis à Brest, EcoTree (<http://www.ecotree.fr>) est une start-up qui depuis 2016 propose d'investir dans les arbres. Un geste en faveur de l'environnement, mais un placement aussi, « une sorte de livret vert », explique son président et co-fondateur, Théophane Le Méné dont le rendement tourne autour de 2 % par an.



En achetant un arbre pour 15 € à 30 € à peine, particuliers ou entreprises ont ainsi la possibilité de participer au reboisement de la forêt française. L'arbre prend ensuite de la valeur et au moment de sa coupe qui intervient entre 25 et 60 ans en fonction des essences, après sa plantation, l'intégralité de la vente du bois est reversée à l'investisseur, ce qui correspond au rendement de 2% annuels annoncés.

Comment ça marche ?



Tout ce qui touche à l'achat des espaces forestiers, leurs gestions, la plantation des arbres, leurs commercialisations une fois coupés, est assuré par EcoTree. « On assure la maîtrise de la chaîne de bout en bout, assure son président. Tous ceux qui ont financé la plantation d'un arbre peuvent, depuis la plateforme et à travers son espace client voir la géolocalisation, les dépenses auxquelles ils ont contribué, toutes les actions réalisées pour la préservation, le maintien et la gestion des forêts. » Ainsi, en 2019, la plateforme affichait 36 119 arbres plantés, 777 485 kg de CO₂ absorbés ce qui correspond à 1 554 vols Paris/New-York. Et le compteur n'est pas prêt de s'arrêter de tourner.

1.2 Présentation du projet

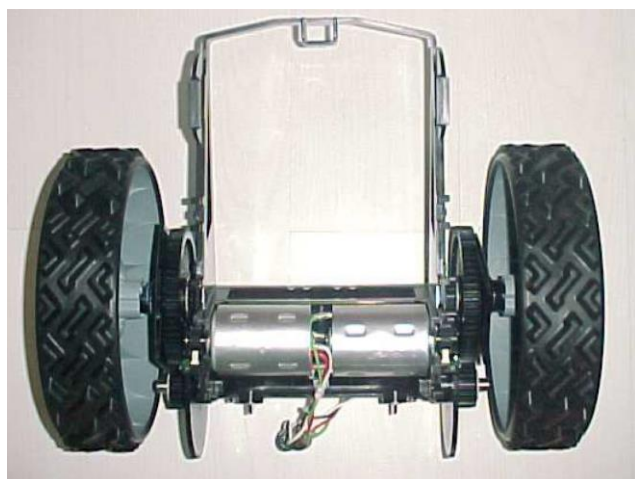
Afin de répondre aux enjeux de développement de son parc forestier déployé sur le territoire national de métropole, la société Ecotree souhaite évaluer la faisabilité d'un projet de développement d'un ROBIOT ou robot de mesure automatisée de l'état d'un parc forestier.

Dans cette optique, la présente consultation est établie aux fins de développer un logiciel de simulation des performances d'un robot autonome en capacité de réaliser des mesures d'états d'un parc forestier. Ledit robot autonome dit ROBIOT doit pouvoir être déployé par un opérateur sur un terrain forestier donc la société assure la gestion. L'optique est d'optimiser le temps de présence d'un ingénieur forestier lors de l'inspection d'une parcelle.

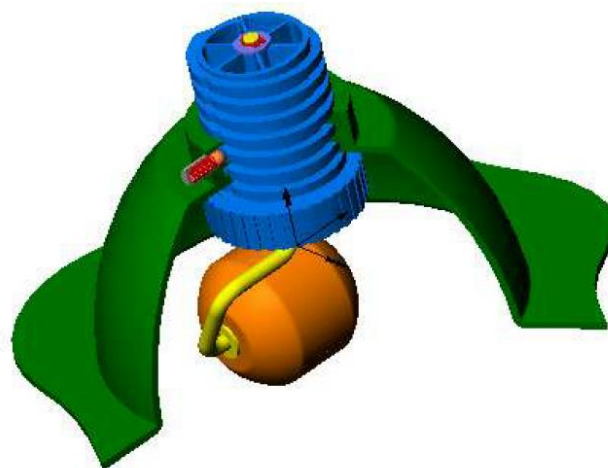
1.3 Fonctionnement de la partie opérative ROBIOT

Le robot autonome objet de l'étude est munie de deux roues motrices arrières et d'une roue « folle » à l'avant tournant librement sur elle-même de type « jockey ».

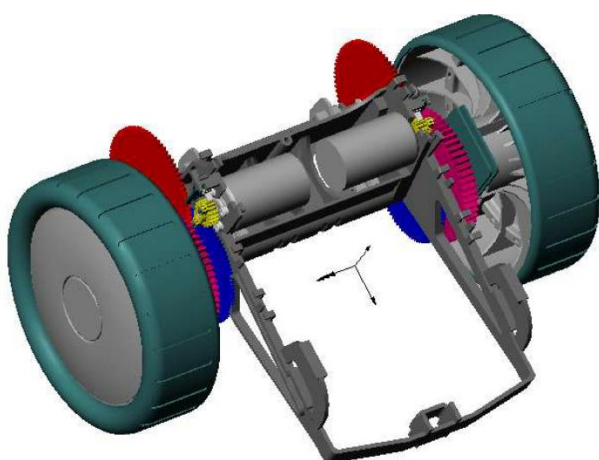
Les deux roues motrices arrières sont toutes les deux motorisées de manière indépendante ce qui permet de contrôler le mouvement linéaire et rotatif du robot. Les roues motrices sont montées dans un berceau dont la position par rapport au châssis est réglable de manière à adapter la hauteur de fonctionnement du robot au terrain de déploiement. La position de la roue jockey par rapport au châssis est également réglable en association avec le réglage du berceau.



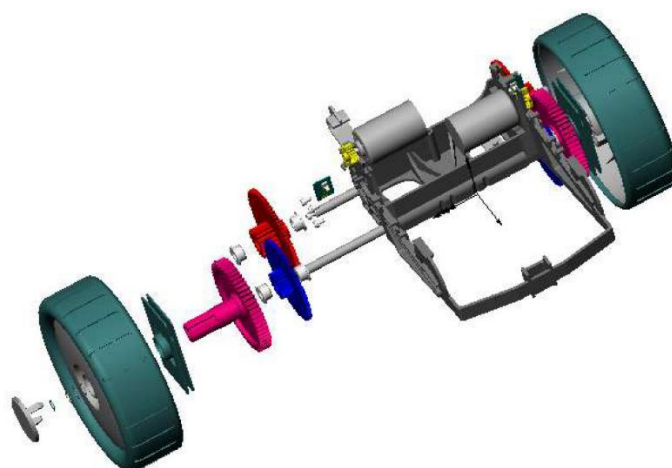
PROTOTYPE DE CHASSIS MOTORISE



MODELISATION DE ROUE JOCKEY



MODELISATION DE CHASSIS MOTORISE

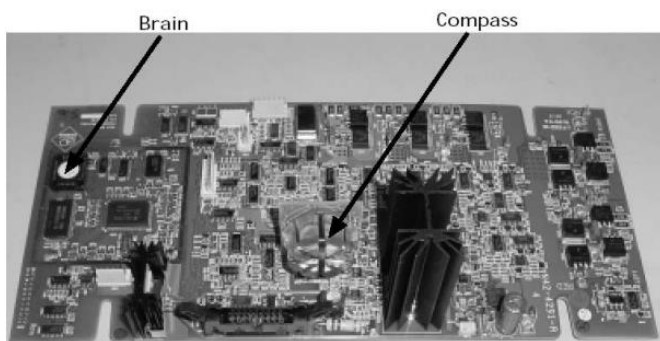


MODELISATION EN VUE ECLATEE DE CHASSIS MOTORISE

1.4 Fonctionnement de la partie commande

Le robot possède une carte mère regroupant les différentes fonctions de puissance et de commande. Les seuls éléments décentralisés de cette carte mère sont les différents capteurs qui renseignent la partie commande sur l'état du système ou les événements qui se produisent pendant le fonctionnement.

Un boîtier de commande permet de configurer les paramètres de mesure et de déplacer le robot soit pour une prise de mesure manuelle ou pour déplacer le robot sur la zone de travail.



PROTOTYPE DE CARTE MERE



PROTOTYPE DE BOITIER DE COMMANDE

1.5 Documents applicables et de référence

Au titre du présent projet de développement, il est fourni les documents et références suivants :

- ✓ [CCTP] – Cahier des Clauses Techniques Particulières – ROBIOT SIMU – V1R0 – 15 Juin 2020
- ✓ [QL-DEV] – Normes de développement C++ - V1R0 – Juin 2020
- ✓ [CNCC] – Outil de métrique qualité du code - <https://sourceforge.net/projects/cncc/>
- ✓ [TRELLO] – Outil de pilotage projet client – <https://www.trello.com>
- ✓ [IDE] – Environnement de développement intégré - <https://visualstudio.microsoft.com/fr/vs/community/>
- ✓ [PROFIL] – Découvrir les
- ✓ [GITHUB] – Environnement d'intégration continue - <https://github.com/>

2. Expression du besoin

2.1 Positionnement du simulateur

Le projet de développement objet de la présente consultation doit fournir une solution logicielle de simulation des capacités du ROBIOT. Ledit projet de développement est nommé **ROBIOT SIMU**.

2.2 Objectifs et besoins logiciels

La société Ecotree bénéficie d'une croissance solide depuis sa création. Dans le cadre de son développement, elle souhaite disposer d'un outil performant de gestion de son parc forestier sur le territoire national. Dans cette optique, l'assistance d'un robot autonome de mesure de l'état du parc est un outil indispensable à la pérennité du modèle de développement de la société.

La société a d'ores et déjà engagé des travaux de prototypage mécanique du ROBIOT. Cependant, afin d'optimiser le dimensionnement et les capacités du robot et valider l'implémentation de l'algorithme de DIJKSTRA pour l'optimisation du parcours du ROBIOT, il est attendu au titre de la présente consultation de mettre en œuvre un simulateur du code du ROBIOT.

2.3 Exigences fonctionnelles

La solution ROBIOT SIMU à mettre en œuvre doit permettre la validation du code de pilotage du déplacement autonome du ROBIOT et l'identification des besoins d'autonomie.

Le principe de déplacement du robot repose sur l'algorithme de DIJKSTRA pour résoudre le problème du plus court chemin (https://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra).

Le cœur du système à développer est donc le Contrôleur de déplacement et de mesure du ROBIOT.

Le contrôleur logiciel doit disposer dans son environnement des données issues des capteurs et des interfaces de commande des actionneurs. Les composants logiciels permettant de simuler ces éléments dans le ROBIOT SIMU sont les suivants :

- ✓ CMesure - Pilote d'outil de mesure forestier
- ✓ CCompas – Capteur de géolocalisation du ROBIOT
- ✓ CBatterie – Capteur de niveau de charge de la batterie du ROBIOT
- ✓ CMoteur – Pilote du couple de moteur du châssis
- ✓ CCapteur – Capteur de signaux balise (Bordures de terrain, Obstacles, Arbres)
- ✓ CCommande – Boîtier de commande (Fourniture des listes d'arbres à contrôler)

2.3.1 Partie commande

Sur la partie commande du ROBIOT, le simulateur doit utiliser deux fichiers d'entrée pour l'exécution d'un parcours simulé : la cartographie du terrain et la liste des arbres à contrôler.

Le premier fichier permet de simuler le comportement du système de Capteur de détection et définition de cartographie terrain. Il s'agit d'une technologie radiofréquence permettant d'obtenir la position de balises de bordures de terrain et l'identifiant des balises d'attachement. Le système donne aussi la position des obstacles et des arbres avec leur identification. Un exemple de donnée est fourni en annexe du présent document.

Le second fichier permet de simuler le comportement du boîtier de Commande par lequel l'opérateur sélectionne les arbres objet d'une mesure de contrôle d'état. Un exemple de donnée est fourni en annexe du présent document.

Le Contrôleur du ROBIOT repose sur une carte mère d'architecture ARM dont la consommation moyenne en fonctionnement est de 12.5 Watts.

2.3.2 Partie opérative

Sur la partie opérative, le Contrôleur peut piloter le Moteur via l'interface de ce dernier pour obtenir le déplacement du ROBIOT. Le Moteur permet le déplacement du ROBIOT à une vitesse 0.42 m.s^{-1} sur un terrain de pente maximale de 10%.

Une première hypothèse de travail pour le simulateur permet de considérer que le travail du moteur sur les roues induit un déplacement proportionnel et donc une modification équivalente des coordonnées du ROBIOT mesuré par le Compas.

Une seconde hypothèse permet de considérer une consommation moyenne du moteur stable indépendamment des conditions du terrain (pente et qualité de surface). La puissance moyenne demandée par le moteur d'une roue est de 14,0 Watts.

2.3.3 Partie mesure

Sur la partie outil de mesure forestière, le ROBIOT effectue ses mesures sur un arbre en étant immobile afin de déployer les outils et effectuer les mesures. Le système de mesure réalise une procédure identique pour tous les arbres qu'il doit contrôler. Le cycle de mesures prend une durée moyenne de cinq minutes par arbre et la puissance moyenne consommée par le système de mesure est de 30,0 Watts.

2.4 Exigences non-fonctionnelles

L'objectif du simulateur est notamment l'identification des besoins d'autonomie du ROBIOT. Le développement doit offrir deux informations de mesures que sont les triplets temps de parcours/nombre de mesures/consommation et les profils d'utilisation des ressources processeur et mémoire.

Bien que le ROBIOT repose sur une architecture ARM, le simulateur est réalisé pour des cibles d'architecture IA64. L'analyse en corrélation des données de consommation par parcours en rapport à l'usage des ressources doit permettre de valider la représentativité des simulations.

De plus, dans le contexte du projet la qualité peut s'aborder suivant deux perspectives :

- ✓ La qualité externe, validée via les tests unitaires selon le Framework Google Test ;
- ✓ La qualité interne et la maintenabilité, vérifiées via des revues de code et des contrôles automatiques du code.

Ces éléments de contrôle doivent faire partie de la définition du « Done » au sens de l'approche agile retenu pour cette consultation (cf. section « Cadre de prestation ». Le non-respect de la qualité interne du produit, que ce soit au niveau du code ou de l'architecture est source de dette technique et donc d'augmentation du temps d'intervention des développeurs.

3. Cadre de prestation

3.1 Cadre d'exécution des travaux

Le projet est conduit sur la période courant du **Lundi 15 Juin 2020** au **Mardi 23 Juin 2020**. La date limite de dépôt des livrables est arrêté au **Mardi 23 Juin 2020 à minuit (Heure de Paris)**.

Le projet est à réaliser en trinôme dont la constitution est laissée libre. Les binômes sont autorisés si nécessaire, les équipes projet de plus de trois personnes sont refusées.

La réalisation des travaux est assurée en heures et jours ouvrées dans le cadre de la planification des heures de formation, à savoir **un total de 24 heures**.

Le projet est donc prévu en équivalent temps plein pour **une charge de 9 j/h**.

3.2 Qualité logicielle

Le développement logiciel doit être réalisé en **langage C/C++** avec une **interface utilisateur en mode texte**.

L'environnement de développement intégré de type **Microsoft® Visual Studio** sera utilisé pour le projet.

L'intégralité des développements doit faire bénéficier d'un cadre d'intégration continue minimal par l'utilisation d'un référentiel de gestion de code source de type **GitHub**. La gestion du code doit produire trois trains de versions du code :

- ✓ **BUILD** – Ensemble des codes sources en cours de développement et valide pour compilation
- ✓ **TEST** – Ensemble des codes sources figés dans une version pour tests unitaires et de validation
- ✓ **PROD** – Ensemble des codes sources bons pour livraison (partielle ou finale) au client

Les **tests unitaires** sont à mettre en œuvre via le Framework **GOOGLE TEST**.

Les **tests de validation client** sont basés sur trois jeux de données d'entrées fournis par le client.

3.3 Livrables de prestation

Au titre de la présente consultation, il est attendu la fourniture de livrables suivants :

- ✓ L'ensemble des codes sources dans les trois trains de version (cf. paragraphe précédent)
- ✓ Le rapport de performance d'exécution de l'application issu de l'outil « PROFILER » de MS® Visual Studio
- ✓ Le rapport de métrique qualité du code issu de l'outil CNCC (<https://sourceforge.net/projects/cncc/>)

La mise à disposition des livrables est **acceptée uniquement au travers du dépôt GITHUB** et l'envoi d'un courriel à l'adresse contact@tanooki.fr au plus tard le **Mardi 23 Juin à minuit** et comportant les éléments suivants :

- ✓ OBJET = [INSA] Livraison ROBIOT SIMU – NOM1 – NOM2 – NOM3
- ✓ CONTENU = Adresse URL du dépôt GITHUB de l'équipe projet

IMPORTANT – Le dépôt GIT doit héberger l'intégralité des livrables du projet tels que susmentionnés.

IMPORTANT – Les droits de lecture devront être ouvert pour permettre un accès au dépôt

IMPORTANT – L'heure de dépôt des fichiers sur la plateforme GITHUB fait autorité pour la limite de livraison

3.4 La gestion de projet / Planification

Le projet offre un cadre particulier d'exécution par les contraintes pédagogiques et la nature exploratoire de son contenu. Par conséquent, une approche agile du développement logiciel est à mettre en œuvre. En effet, la présence du client (le formateur) est assurée en proximité de l'équipe de développement pendant la durée du projet. De plus, le cycle de développement court et le cadre de l'étude de faisabilité ne faciliterait pas un mode de gestion de projet classique.

Pour permettre le suivi du projet par les différentes parties prenantes, il est demandé de mettre en place une gestion de projet agile portée par un outil partagé de type Kanban. Cette contrainte du projet est établie dans une optique pédagogique, afin d'explorer les impacts d'une telle gestion sur l'assurance qualité logicielle.

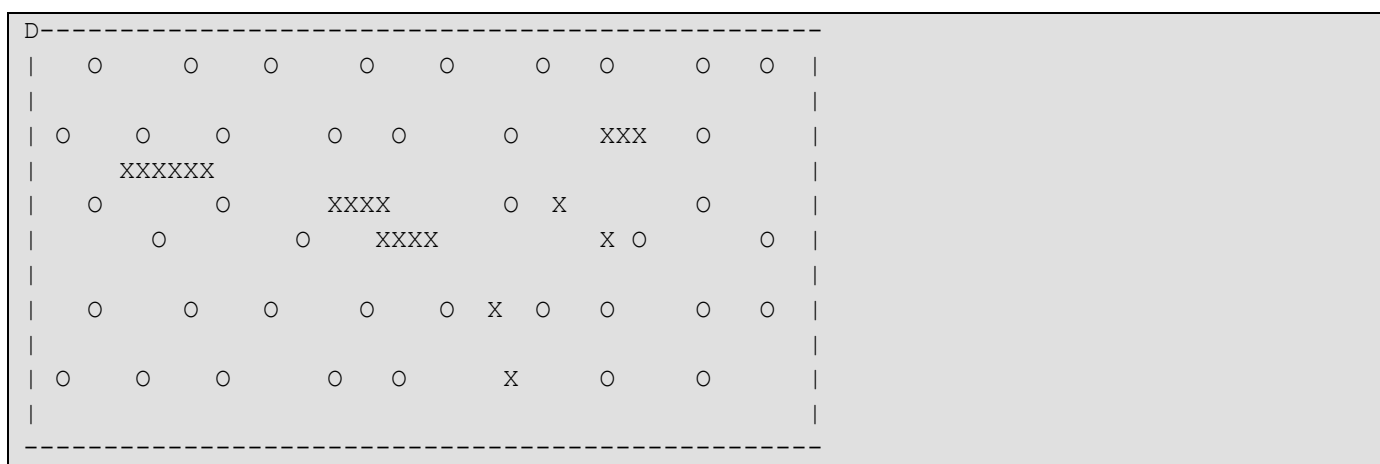
Afin de cadrer la démarche et faciliter le travail des équipes, il est demandé d'utiliser l'outil TRELLO (<http://www.trello.fr>) afin de mise en œuvre de ce pilotage agile.

4. Annexes

4.1 Exemple de données de simulation – CARTOGRAPHIE TERRAIN

Légende des données :

- ✓ La lettre D indique le point de départ du ROBIOT pour le parcours simulé
- ✓ La lettre O indique l'emplacement d'un arbre
- ✓ La lettre X indique l'emplacement d'un obstacle à contourner
- ✓ Les symboles | et – indique les bordures du terrain forestier



4.2 Exemple de données de simulation – LISTE DE CONTROLE

Le fichier indique les coordonnées des arbres à contrôler par des couples « X Y ».

Les cartographies indiquent des coordonnées relatives au point de départ du ROBIOT.

L'axe des abscisses est horizontal et orienté vers la droite.

L'axe des ordonnées est vertical et orienté vers le bas.

```
4 1
4 5
12 5
21 8
32 8
36 10
```