

UNIVERSITÉ PAUL SABATIER



MASTER INTELLIGENCE ARTIFICIELLE ET
RECONNAISSANCE DES FORMES
MASTER ROBOTIQUE : DÉCISION ET COMMANDE

Plan de Développement Qualité

Navigation Autonome de Robot Mobile

Auteur :

Thibaut AGHNATIOS
Marine BOUCHET
Bruno DATO
Tristan KLEMPA
Thibault LAGOUTE

Tuteur :

Frédéric ERASLE
Michaël LAUER
Michel TAIX

Equipe Navigation

Plan Qualité

REF

Table des matières

1	[AMODIFIER] Introduction	4
2	Présentation du projet	4
2.1	[OK] Contexte	4
2.1.1	Master IARF et RODECO	4
2.1.2	Contexte du projet	5
2.1.3	Entrées	5
2.1.4	Sorties	5
2.1.5	Limites	5
2.2	[OK] Parties-prenantes	6
2.2.1	MOE	6
2.2.2	MOA	6
2.2.3	Intervenants externes	6
2.3	[ACOMPLETER] Contraintes	6
3	Organisation du projet	6
3.1	[OK] Cycle de développement	6
3.2	[AMODIFIER] Sprints	7
4	[OK] Arbre produit	7
5	[AFAIRE] RoadMap	8
6	Planning	8
6.1	[OK] Lôt de tâches	8
6.2	[OK] Planning général	8
6.3	[ACONFIRMER] Livrables	9
6.3.1	Pour le client	9
6.3.2	Pour les intervenants externes	9
6.4	[OK] Planning du W1	9
6.5	[OK] Conclusion du projet	9
7	Assurance qualité	10
7.1	[AFAIRE] Organisation interne	10
7.2	[AMODIFIER] Organisation externe	10
7.3	[AMODIFIER] Outils et Standards	10
7.3.1	Développement	10
7.3.2	Documentation	11
7.3.3	Espace de travail	11
7.3.4	Entête des fichiers	11

7.3.5	Nommage	12
8	[OK] Analyse des risques	12
9	[AFAIRE] Validation	12

1 [AMODIFIER] Introduction

La structure et le contenu de ce Plan Qualité de Projet ont été élaborés dans le cadre du projet « Navigation autonome de robots mobiles » proposé aux étudiants de deuxième année du Master Intelligence Artificielle et Reconnaissance des Formes » de l'Université Paul Sabatier à Toulouse.

Il a pour objectif la définition et la description des différentes dispositions à mettre en œuvre pour un développement optimal du projet afin d'en assurer la qualité et d'atteindre les résultats attendus. Plus précisément, sont déterminés, d'un commun accord :

- l'organisation globale du projet
- le plan de gestion et de développement du projet
- les droits et les devoirs de chaque partie prenante
- la répartition des responsabilités entre les organismes dans la structure
- les plans de développement et de gestion du projet
- les outils qui seront adoptés

Après acceptation par le client, ce plan deviendra le document contractuel applicable en matière de gestion et d'assurance qualité entre le Titulaire et le Client durant la durée totale du projet. La responsable qualité s'assurera qu'il est effectivement appliqué. Il ne pourra subir aucune modification sans accord préalable du Client. En cas de divergence entre les exigences et le plan qualité, les exigences s'appliqueront en priorité.

2 Présentation du projet

2.1 [OK] Contexte

2.1.1 Master IARF et RODECO

Le master « Intelligence Artificielle et Reconnaissance des Formes » (IARF) a comme objectif de former des professionnels de haut niveau capables de concevoir des solutions à des problèmes complexes utilisant des méthodes avancées de représentation et de traitement de l'information, faisant appel à des techniques d'intelligence artificielle (IA) et de reconnaissance des formes (RF) et d'apprentissage automatique, appliqués notamment au traitement d'images et à la robotique.

Le master « Robotique : décision et commande » (RODECO) a pour vocation de _ des connaissances dans le domaine de l'automatique par des enseignements avancés autour de la robotique, de l'informatique et de la commande des systèmes. Ces compétences permettent d'aborder des problématiques très actuelles comme la robotique industrielle haute performance où les aspects commande sont fondamentaux et la robotique de service où la décision et la perception tiennent une place essentielle. Suivant ce raisonnement, deux blocs de spécialisation sont proposés en M2 :

Robotique et décision qui propose un renforcement des aspects « informatique » (intelligence artificielle, reconnaissance des formes, dialogue homme/machine), vision par ordinateur et robotique mobile. Cette spécialisation donne les compétences nécessaires pour appréhender le domaine de la robotique de service ;

Robotique et commande qui se focalise sur le développement et l'implantation de commandes avancées pour la robotique. Cette spécialisation donne donc les compétences nécessaires pour élaborer des solutions évoluées de contrôle/commande pour la réalisation de tâches robotique haute performance.

Au cours de cette deuxième année, les étudiants acquièrent une double compétence en Automatique et Informatique et les capacités requises pour modéliser, analyser, concevoir et réaliser des systèmes automatiques complexes, autonomes et/ou embarqués où sont impliqués la perception (capteurs), l'analyse (traitement de signal, audio, image, vidéo), le raisonnement et la décision (incertitude, reconnaissance de formes, contraintes) et de l'action (commande, robotique).

2.1.2 Contexte du projet

Le projet de Master 2 permet de mettre en commun et en pratique les connaissances acquises dans ces trois parcours dans un but commun. En l'occurrence, sur le projet « Navigation autonome de robots mobiles », Tristan et Marine font parti du parcours IARF, Thibaut et Bruno sont issus de la spécialité Décision de RODECO et enfin, Thibault, de la spécialité Commande.

Il se déroule tout le long de l'année, en alternance avec les blocs de cours B par tranche, nommés W, de 3, 4 et 4 semaines.

B1	W1	B2	W2	B3	W3
5 sem.	3 sem. 17/10 - 14/11	5 sem.	4 sem. 02/01 - 27/01	5 sem.	4 sem. 01/03 - ??/03

2.1.3 Entrées

- ☒ Cahier des charges
- ☒ Documentations

2.1.4 Sorties

L'objectif du projet est de répondre à un cahier des charges divisé en 4 étapes incrémentales. Le projet étant enmené à être réutilisé par la suite pour d'autres ... , la qualité est primordiale.

- ☐ Un code propre, fonctionnel et surtout facilement réutilisable
- ☐ Documentations
- ☐ Robot opérationnel qui ...
- ☐ Manuel Utilisateur

2.1.5 Limites

- Incrémentation des solutions : difficulté d'anticiper les solutions et problèmes des étapes suivantes
- Cours de vision 2D et de navigations en B2 et vision 3D en B3

2.2 [OK] Parties-prenantes

2.2.1 MOE

Thibaut AGHNATIOS
thibaut.agnatios@laposte.net
Spécialité Décision

Tristan KLEMPKA
klempka.tristan@gmail.com
Spécialité IARF

Marine BOUCHET
bouchetmarinee@gmail.com
Spécialité IARF

Thibault LAGOUTE
lagoute.31@gmail.com
Spécialité Commande

Bruno DATO
bruno.dato.meneses@gmail.com
Spécialité Décision

2.2.2 MOA

Frédéric LERASLE
lerasle@laas.fr
Equipe RAP, LAAS - CNRS

Michel TAIX
taix@laas.fr
Equipe GEPETO, LAAS - CNRS

Michel LAUER
michael.lauer@laas.fr
Equipe TSF, LAAS - CNRS

2.2.3 Intervenants externes

Julien VANDERSTRAETEN
julien.vanderstraeten.ups@gmail.com
Coach

Cyril BRIAND
briand@laas.fr
Coach

2.3 [ACOMPLETER] Contraintes

La réalisation du projet est régie par certaines contraintes citées ci-dessous :

- Robot TurtleBot
- Planning du Master et ses jalons
- Connaissance partiele + entre les membres

3 Organisation du projet

3.1 [OK] Cycle de développement

Le développement du projet se fera selon Scrum, une méthode Agile dédiée à la gestion de projet. Cette méthode, basée sur les stratégies itératives et incrémentales, permettra de produire, à la fin de chaque « sprint », un résultat achevé et validé, pour avoir une version fonctionnelle à tout moment. Cette démarche correspond bien au cahier des charges où les étapes sont itératives et incrémentales.

Pour chaque sprint, la période initiale permettra de faire le point sur les éléments techniques du départ du projet et d'établir une liste des points à préciser ou à compléter, les charges de travail et le calendrier associé. Une nouvelle version du Plan Qualité et du Plan Développement adapté à l'étape en cours sera donc produite. Chaque sprint donne lieu à une phase de codage et une phase de tests.

Scrum se différencie des autres méthodes de développement par ses avantages qui font de ce procédé une réponse à certains problèmes fréquemment rencontrés dans le développement logiciel. Pour éviter l'effet tunnel, la communication restera permanente entre les membres de l'équipe mais aussi entre l'équipe et le client, soit une meilleure coopération à l'intérieur de l'équipe Scrum.

SCHEMA A FAIRE

3.2 [AMODIFIER] Sprints

Le projet est divisé en 4 sprints :

Etape 1 Réaliser une tâche robotique permettant au robot d'atteindre une balle de couleur (détecter une balle de couleur, localisation dans le repère courant du robot, aller à la balle).

Etape 2 Réaliser une tâche de navigation dans un environnement supposé connu et sans obstacles pour atteindre un amers final en se déplaçant d'amer en amers (2D - QR code) afin de ne pas se perdre et en optimisant un critère (distance, temps, commande,...).

Etape 3 Réaliser une tâche de navigation dans un environnement supposé connu et sans obstacles inconnus pour atteindre un amers final en se déplaçant d'amer en amers (3D) afin de ne pas se perdre et en optimisant un critère (distance, temps, commande,...).

Etape 4 Réaliser une tâche de balayage de grâce à une modélisation par carte laser.

A chaque étape, une étude est nécessaire et demandée.

EX : Analyse des fonctionnalités disponibles sur le TurtleBot via ROS en terme de capteurs (perception et localisation) et de commande (navigation).
OU études biblio.

+ ETAPE 4 FACULTATIVE

SCHEMA

4 [OK] Arbre produit

Un arbre produit, ou Product Breakdown Structure donne une liste exhaustive des différents livrables du projet, de manière hiérarchisée. Il permet d'avoir une vue claire des différentes fonctionnalités à développer, de décrire l'architecture logicielle et matérielle du produit à livrer, et d'en prévoir les coûts. Dans le cadre de notre projet de navigation, l'aspect budgétaire ne sera pas abordé puisqu'il s'agit d'un projet dans le cadre de la formation. La figure suivante présente l'arbre produit du projet de grande envergure. Les parties encadrées

correspondent aux nœuds de l'arbre, c'est à dire les groupes de fonctions à développer, tandis que les parties non encadrées correspondent aux feuilles, autrement dit aux fonctions elles-mêmes. L'arbre produit de notre projet de navigation est décomposé en cinq parties :

Simulateur ROS : concerne les différents éléments à utiliser sous ce simulateur, à savoir la modélisation de l'environnement, les différents obstacles, le calcul des trajectoires de notre robot et l'intégration des blocs au cours des différentes étapes.

Perception : décrit l'ensemble des fonctionnalités permettant l'identification à partir d'une image. On distingue ainsi 2 groupes : la perception de l'environnement et la détection des obstacles. La détection des obstacles tient compte des différentes méthodes utilisées au cours de ce projet : le nuage de points, l'étude laser, les différentes méthodes de segmentation, etc. . .

Localisation : décrit les fonctions nécessaires à la navigation autonome du robot. Il devra permettre au robot de calculer sa position et déterminer son orientation grâce à des repères.

Décision : décrit les fonctions nécessaires aux différentes décisions que le robot doit prendre comme pour la génération de la trajectoire (en choisissant la trajectoire la plus courte et la plus rapide à prendre selon l'orientation du robot) et les décisions pour les différents événements possibles.

Action : Concerne toutes les fonctions correspondant au domaine de la commande robotisée permettant le respect des consignes (atteindre une balle de couleur, un amer, . . .), la réalisation des mouvements, et les contraintes cinématiques. On parlera aussi de l'odométrie.

ARBRE

5 [AFAIRE] RoadMap

6 Planning

6.1 [OK] Lôt de tâches

Un lot de tâches, ou Work Breakdown Structure, est une décomposition hiérarchique des tâches à effectuer dans un projet. Il permet à la fois d'avoir une représentation visuelle du travail restant à réaliser, mais aussi d'évaluer rapidement le budget temps par rapport à l'argent, à chacune de ses tâches.

La figure suivante représente nos lots de tâches pour l'ensemble du projet, nous en avons identifié six.

DIAGRAMME

6.2 [OK] Planning général

GANTT OK

6.3 [ACONFIRMER] Livrables

6.3.1 Pour le client

W1 du 17/10 au 14/11 :

- Plan Développement V1 : 4/11
- Plan Qualité V1 : 3/11
- Code de l'étape 1 et Manuel Utilisateur correspondant : 8/11

W2 du 02/01 au 27/01 :

- Etat de l'Art de l'étape 2 : à définir avant le W2
- Plan Qualité V2 : 02/03
- Backlog de l'étape 2 : 02/03
- Plan Développement V2 : à définir avant le W2
- Code V2 et Manuel Utilisateur : à définir avant le W2
- Cahier de recettes V2 : à définir lors avant le W2
- Plan Qualité V3 : à définir avant l'étape 3
- Backlog de l'étape 3 : à définir avant l'étape 3
- Plan Développement V3 : à définir avant l'étape 3

W3 du 01/03 au ??/03 :

- Etat de l'Art de l'étape 3 : à définir lors avant l'étape 3
- Code V3 et Manuel Utilisateur : à définir avant le W3
- Cahier de recettes V3 : à définir avant le W3
- Backlog : à définir avant l'étape 4
- Plan Développement V4 : à définir avant l'étape 4
- Plan Qualité V4 : à définir avant l'étape 4
- Code V4 et Manuel Utilisateur : à définir avant l'étape 4
- Bilan de projet : à définir avant le W3
- Cahier de recettes : à définir avant le W3

Tout au long du projet :

- Compte-rendus de réunions : ?

6.3.2 Pour les intervenants externes

- Compte-rendus de réunions
- Plan Développement
- Plan Qualité

6.4 [OK] Planning du W1

GANTT OK

6.5 [OK] Conclusion du projet

Ce plan qualité vise à assurer que les dispositions prises par l'équipe pour obtenir la qualité du logiciel définie en accord avec les clients soient respectées. Avec le même objectif, le responsable qualité, soutenu par toute l'équipe, sera en charge de vérifier que les engagements pris dans le présent document auront été appliqués tout au long de l'avancement de ce projet.

Les membres de l'équipe projet sont tenus de se conformer aux dispositions décrites dans le plan d'assurance et de contrôle qualité. Le non-respect des prescriptions du Plan Qualité constaté donne lieu à un plan d'action curatif pour corriger les effets du dysfonctionnement et éventuellement à un plan d'action correctif pour éviter que celui-ci ne se reproduise. Ce dernier plan peut entraîner une modification du plan qualité. L'utilisation de ce plan doit permettre un total succès du projet :

Succès du produit : avoir un produit final qui satisfait les besoins des utilisateurs, qui a le niveau de qualité requis (tests, code propre et commenté) et qui puisse être évolutif (générique) et réutilisable (documentations)

Succès de la démarche : les demandes du client satisfaites dans les délais, les méthodes établies et le planning bien respectés

Le projet et les résultats obtenus seront décrits dans le bilan et exprimés lors de la présentation orale et démo publique.

7 Assurance qualité

7.1 [AFAIRE] Organisation interne

7.2 [AMODIFIER] Organisation externe

La communication avec les clients se fera premièrement par courrier électronique. Les clients pourront alors répondre à toute l'équipe via l'adresse mail du groupe mise à leur disposition. Deuxièmement, des réunions hebdomadaires seront également prévues afin de tenir informés les intervenants sur l'avancement du projet. Des réunions supplémentaires pourront également être organisées si cela s'avère nécessaire. Chaque réunion sera prévue au moins 2 jours à l'avance et donnera lieu à un compte rendu de réunion. Ces deux comptes rendus seront envoyés par courriel, dont l'objet sera identifié VOIR PARTIE, et ils seront rédigés à partir des formulaires fournis en Annexe. Ces documents seront soumis à l'approbation dans les trois jours. De plus, toutes les informations relatives au projet, quelle que soit leur nature seront accessibles immédiatement dans nos espaces de travail VOIR PARTIE.

7.3 [AMODIFIER] Outils et Standards

7.3.1 Développement

- * Systèmes d'exploitation : Ubuntu 14.04 LTS

- * ROS

Git : Logiciel libre de gestion de versions. Nous l'utilisons pour la gestion de nos sources mais aussi pour gérer la rédaction collaborative de nos documents.

Github : Hébergeur de notre dépôt Git. L'interface Web fourni également un accès à des outils de gestion du code. Une liste d'Issues qui sera utilisé pour discuter des modifications effectuer dans les codes et un système de

Pull Request qui nous permet de valider et d'intégrer des modifications.
Un wiki est également accessible via cette interface.

7.3.2 Documentation

* Langage de modélisation : UML

Google Drive : Système de stockage et de partage en ligne de fichiers. Il permet d'archiver et d'avoir accès aux versions finales des documents rédigés et utilisés durant le projet.

Cacoo : Service en ligne de création et de partage de diagramme.

LaTeX - TexMaker : LaTeX est un langage de rédaction de document. Il permet la rédaction de document et l'intégration de plusieurs parties facilement. Nous avons décidé de choisir une distribution LaTeX commune pour s'assurer que les documents produits seront homogènes (encodage, caractères spéciaux... etc)

7.3.3 Espace de travail

GanttProject :

Trello : Outil de gestion de projet en mode « Tableau de tâches ». Il nous permet d'organiser la distribution et la réalisation des tâches. Il sert également comme vecteur d'information sur la gestion du projet en général (liens des outils, documents importants... etc.).

Google Mail : Service de messagerie électronique. Il est utilisé pour l'ensemble de nos échanges internes ou externes au projet.

Google Agenda : Service en ligne d'agenda. Utilisé pour la prise de rendez-vous et la mise en place des réunions internes et externes au groupe de projet.

Il est important que le code soit suffisamment découpé. La taille des fichiers ne devrait pas excéder 1000 lignes et chaque fonction (ou méthode) ne devrait pas dépasser la centaine de lignes au grand maximum. Les headers et les fichiers source séparent les déclarations des définitions. L'indentation doit être strictement respectée et doit correspondre à un espacement de 4 caractères. La taille des lignes ne doit pas dépasser 80 caractères.

7.3.4 Entête des fichiers

L'entête de chaque fichier source doit être complétée afin d'expliquer leur comportement. Elle doit être mise à jour au fur et à mesure des évolutions du fichier qu'elle décrit. Chaque entête doit permettre de comprendre facilement et rapidement les fonctionnalités codées. Voici une liste non exhaustive des éléments à renseigner :

- Méthode/Fonction
 - D : Descriptif
 - A : Auteur(s)
 - E : Description des paramètres
 - S : Donnée(s) renvoyée(s)

- R : Donnée renvoyée
- F : Exceptions et/ou code d'erreurs renvoyées
- Classe
 - Descriptif
 - Auteur(s)

Il est recommandé de commenter au maximum le code afin de faciliter la phase de développement du projet.

7.3.5 Nommage

Les fichiers suivent le nommage de leur classe respective. Pour les fichiers C++ c'est le suffixe cpp qui sera utilisé. Pour les headers C++ le suffixe hpp sera utilisé.

Classes :

- Première lettre en majuscule
- Mélange de minuscule, majuscule.
- Première lettre de chaque mot en majuscule
- Donner des noms simples et descriptifs
- Éviter les acronymes
- N'utiliser que [a-z][A-Z] et [0-9]

Variables :

- Première lettre en minuscule
- Mélange de minuscule, majuscule avec la première lettre de chaque mot en majuscule
- Donner des noms simples et descriptifs
- N'utiliser que [a-z][A-Z] et [0-9]

Constantes :

- Tout en majuscule
- Séparer les mots par des underscore
- Donner des noms simples et descriptifs
- N'utiliser que [A-Z] et [0-9]

8 [OK] Analyse des risques

TABLEAU

9 [AFAIRE] Validation

Equipe Navigation

Plan Qualité

Table des figures

Equipe Navigation

Plan Qualité

ANNEXE

Templates CR

DICO