

## *Road Runner*

# Plan de projet

Machine à laver

Version 1.0

**Laurence de Villers** (DEVL06568908)

**Adam Smith** (SMIA16068807)

**Valentin Debris** (DEBV04089209)

**Martin Desharnais** (DESM21099102)

**Date 2013-09-27**

Le plan de projet donne une vue d'ensemble de projet et détaille les coûts, l'échéancier et la répartition du travail.

### Historique des révisions

Date (aaaa-mm-jj)	Version	Description	Auteur
2013-09-20	0.1	Démarrage du projet	Laurence de Villers Adam Smith Valentin Debris Martin Desharnais
2013-09-27	1.0	Première version complète	Laurence de Villers Adam Smith Valentin Debris Martin Desharnais

### Abréviations/acronymes

Abré./Acro.	Définition
PM	Project Manager (gérant de projet)
WC	Work Team (équipe de travail)
DO-178	Software Considerations in Airborne Systems and Equipment Certification

---

### Table des matières

#### Plan de projet

1. Introduction .....	4
2. Vue d'ensemble du projet.....	4
2.1. Profil des libertés et contraintes.....	5
2.2. Criticité du projet .....	6
2.3. Références .....	7
2.3.1. Références .....	7
2.3.2. Références normatives .....	8
2.4. Portée.....	8
2.5. Livrables .....	8
3. Organisation du projet .....	9
3.1 Composition de l'équipe du projet .....	9
3.2. Stratégie de contrôle des versions.....	9
3.2.1. Outil de gestion de versions.....	9
3.2.2. Outil de documentation.....	10
3.2.3. Outil de gestion des demandes de modifications.....	10
3.3. Directives de livraison .....	11
4. Ressources matérielles, calendrier et budget.....	12
4.1. Ressources matérielles .....	12
4.2. Calendrier des tâches du projet.....	12
4.3. Budget .....	13
5. Gestion des risques .....	14

---

### 1. Introduction

Dans le plan de projet du logiciel embarqué, développé par **Road Runner**, qui doit être conçu pour l'intégration dans la nouvelle machine à laver d'**ACME**, plusieurs éléments sont présentés afin de bien pouvoir évaluer et comprendre les enjeux, les coûts, les contraintes ainsi que les objectifs liés au projet.

Ces critères ont tous été évalués de manière exhaustive et respectent le plus fidèlement possible l'entente entre les deux parties en plus de respecter le protocole interne et la culture de qualité de **Road Runner**.

Le présent document a donc pour objectif de bien définir et de présenter le projet de manière plus globale. De plus, un profil des libertés, des objectifs et des contraintes a été mis en place afin d'aider tous les intervenants à bien comprendre les difficultés et les enjeux en place pour ce projet. Ensuite, une brève présentation des membres de l'équipe ainsi que du rôle qu'ils occupent au sein de l'entreprise, un survol de notre stratégie de mise en place du contrôle des versions ainsi que des directives de livraison claires et détaillées démontrent que nous possédons l'expertise requise pour entreprendre un tel projet.

En respectant toutes ces procédures et grâce à notre équipe jeune et dynamique, nous sommes persuadés que le logiciel livré sera un succès et que tous les objectifs fixés seront atteints sans problème.

### 2. Vue d'ensemble du projet

Le projet du logiciel embarqué pour les nouvelles machines à laver **ACME** a pour but premier de fournir les fonctionnalités de base de la machine selon les réglages effectués par l'utilisateur.

Entre autres, le logiciel doit :

- gérer le niveau d'eau dans la cuve;
- gérer la température de l'eau;
- contrôler les cycles de lavage;
- s'assurer de la sécurité de l'appareil lors d'événements particuliers (par exemple, ne pas démarrer le cycle de brassage si le niveau de l'eau est élevé et que la porte est ouverte);
- arrêter la machine lors de la détection de bris physiques qui pourraient être dangereux dans la mesure du possible.

Les objectifs principaux sont donc:

- De respecter toutes les fonctionnalités décrites précédemment en mettant l'emphasis sur celles qui assurent la sécurité des biens matériels et des gens ainsi que celles qui sont liées à des boutons physiques afin que toutes les commandes de l'utilisateur soient respectées.
- De ne pas permettre la livraison du logiciel s'il reste des défauts connus étant donné qu'il coûte très cher d'effectuer des correctifs sur des logiciels embarqués de masse.
- De respecter le budget si les contraintes précédentes sont respectées ou de permettre seulement un léger écart du budget dans le cas contraire.
- De livrer le logiciel d'ici trois mois, avec un écart permis maximal de deux semaines. Le logiciel doit donc être livré, idéalement, au début du mois de décembre 2013 ou, au maximum, le 15 décembre 2013.

Il faut donc s'assurer de remplir notre mandat en atteignant tous ces objectifs tout en respectant les contraintes établies.

### 2.1. Profil des libertés et contraintes

Dimension	Objectif	Contrainte	Degré de liberté
<b>Coûts</b>	Respecter le budget établi au départ.	Le logiciel doit être sans défaut connu, car il est trop compliqué et trop cher d'appliquer des correctifs sur tous les appareils livrés, ce qui peut rapidement faire augmenter les coûts en cas de défauts imprévus qui doivent absolument être corrigés.	25 % d'écart acceptable.
<b>Fonctionnalités</b>	Fournir toutes les fonctionnalités demandées.	Les fonctionnalités en lien avec la sécurité des gens et du matériel doivent obligatoirement fonctionner.  Les fonctionnalités liées à des boutons physiques doivent être présentes, car les	Un minimum de 80% des fonctionnalités qui ne sont pas en lien avec la sécurité et qui ne sont pas associées à un bouton physique doivent être présentes dans la version 1.0 du logiciel, ce qui représente environ 90% des

		boutons doivent permettre à l'utilisateur d'accomplir les tâches désirées.	fonctionnalités totales.
<b>Qualité</b>	Qu'aucune défaillance ne se manifeste chez le client.	La version 1.0 ne peut contenir aucun défaut connu, car le coût de correction du logiciel est très élevé étant donné qu'il faut envoyer des gens sur place pour réparer chaque machine à laver.	Aucune, il ne doit pas y avoir de défauts connus.
<b>Échéance</b>	Version 1.0 livrée d'ici trois mois.		Il est possible de livrer jusqu'à deux semaines en retard si certains défauts persistent, car le coût de correction du logiciel est élevé en plus des bris matériels et physiques que ces défauts pourraient causer.
<b>Personnel</b>	Ne pas nécessiter de ressources humaines supplémentaires.	Équipe de quatre personnes.	

Tableau 1.1 - Tableau sommaire des dimensions négociées pour le projet

## 2.2. Criticité du projet

La criticité du projet, selon la norme DO-178, exprime le niveau sécurité qui doit être appliqué pour le projet. Il y a 5 niveaux de criticité décrits comme suit dans le livre *L'Assurance Qualité Logicielle* <sup>1</sup>:

- Niveau A: un dysfonctionnement du logiciel provoquerait ou contribuerait à une condition de perte catastrophique de l'appareil;

<sup>1</sup> April, Alain et Y. Laporte, Claude, *L'assurance qualité logicielle 1*, concepts de base, édition Lavoisier 383 pages

- Niveau B: un dysfonctionnement du logiciel provoquerait ou contribuerait à une condition dangereuse ou un dysfonctionnement sévère et majeur de l'appareil;
- Niveau C: un dysfonctionnement du logiciel provoquerait ou contribuerait à un dysfonctionnement majeur de l'appareil;
- Niveau D: un dysfonctionnement du logiciel provoquerait ou contribuerait à un dysfonctionnement mineur de l'appareil;
- Niveau E: aucun impact sur le fonctionnement de l'appareil ou la charge de travail du pilote.

Le projet de la machine à laver se situe au niveau C, car si le logiciel de la machine à laver a un dysfonctionnement, il pourrait provoquer, selon le pire scénario, une inondation qui causerait une perte financière aux usagers. En général, si la machine à laver a un dysfonctionnement, les dommages seront à un niveau encore plus mineur: les vêtements seront abimés.

La criticité peut aussi être vue selon le principe des cinq dimensions d'un projet logiciel, de Karl Wiegers. Le schéma suivant montre la relation du profil de liberté du projet. Il permet de voir quelles dimensions sont les plus critiques pour le projet. Voici le diagramme de flexibilité du projet.

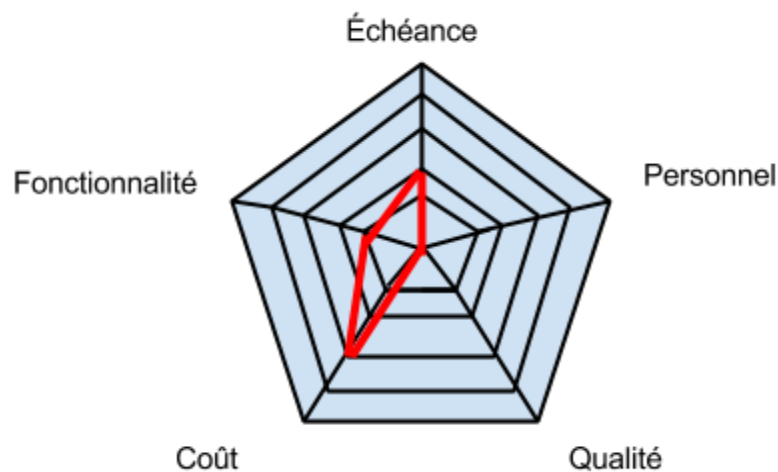


Figure 1.1 Diagramme de flexibilité du projet

## 2.3. Références

### 2.3.1. Références

1. April, Alain et Y. Laporte, Claude, L'assurance qualité logicielle 1, concepts de base, édition Lavoisier 383 pages

---

2. Alain April & Alain Abran, Améliorer la maintenance du logiciel, Loze-Dion, 337 pages

### 2.3.2. Références normatives

- DO-178
- ISO/CEI TR 29110-5-1-1:2012(F)
- ISO/IEC 12207:2008(E)

### 2.4. Portée

La portée du projet du logiciel embarqué de la nouvelle machine à laver d'**ACME** consiste à créer le logiciel qui contrôle les différents éléments de la machine à laver.

Il faut donc que le logiciel contrôle plusieurs éléments important lors des cycles de lavage, comme le niveau et la température de l'eau, les éléments de sécurité pour le matériel et les personnes (ne pas permettre au cycle de brassage de commencer si le niveau de l'eau est élevé et que la porte est ouverte, par exemple), la gestion de quand démarrer et quand terminer les cycles ainsi que la durée de chaque cycle de lavage.

### 2.5. Livrables

Les éléments suivants ont été identifiés comme étant "livrables". Tous ces éléments seront donc remis, tôt ou tard dans le projet, à la compagnie **ACME**.

- Les contrôleurs contenant le logiciel embarqué.
- Le plan de projet.
- Tableau du modèle d'affaires.
- Tableau du code de déontologie.
- Document de spécifications.
- Document de l'utilisateur du logiciel.
- Résultats des tests de validation.
- Registre des corrections.
- Rapport d'analyse et des mesures de recommandations.
- Documentation de l'utilisateur.
- Documentation de la maintenance.
- Formulaire d'enregistrement de réception du logiciel.



### 3. Organisation du projet

#### 3.1 Composition de l'équipe du projet

Rôle	Description	Personne
Gestionnaire de projet (PM)	Le gestionnaire de projet dirige et prend en charge les prises de décisions, la planification du projet, des échéanciers et du budget, gère et supervise les autres membres du personnel dont il ou elle dispose.	Laurence de Villers
Équipe de travail (WT)	Membres de l'équipe qui s'occupent de la production du projet.	Adam Smith Valentin Debris Martin Desharnais

#### 3.2. Stratégie de contrôle des versions

Le service web GitHub<sup>2</sup> est utilisé comme outil de gestion du développement du logiciel. Celui-ci offre, entre autres, les outils suivants :

- hébergement du code source
- documentation collaborative sous forme de Wiki
- système de suivi des demandes de modifications

Le dépôt est disponible exclusivement à l'équipe de projet à l'adresse suivante:  
<https://github.com/authchir/log330-washing-machine-system>

Afin de s'identifier comme membre de l'équipe du projet, un nouvel employé devra se créer un compte GitHub et demander au chef de projet d'être ajouté parmi la liste des collaborateurs. Dès lors, il aura accès au projet à l'aide de l'identifiant de son compte GitHub.

##### 3.2.1. Outil de gestion de versions

Git<sup>3</sup> sera utilisé comme logiciel de gestion des versions décentralisé. Le dépôt central du projet sera disponible par l'outil d'hébergement du code source de Github. Celui-ci contient en tout temps une branche nommée « master » qui contient une version compilable et fonctionnelle du logiciel. Des

---

<sup>2</sup> <https://github.com/>

<sup>3</sup> <http://git-scm.com/>

branches de développement seront créées afin de contenir les modifications qui ne sont pas encore fonctionnelles.

Le choix d'un outil de gestion des versions décentralisé, en opposition à un outil de gestion des versions centralisé a comme avantage de permettre aux différents membres de l'équipe de projet de travailler séparément, d'effectuer des « commits » sur leur ordinateur local et de synchroniser leur projet local avec le dépôt central uniquement lorsque leurs modifications respectent les contraintes imposées par la branche « master » ou toute autre branche de développement.

Afin de donner une idée générale de l'utilisation de l'outil, un exemple d'utilisation est fourni dans cette section.

```
~ $ git clone https://github.com/authchir/log330-washing-machine-system
log330

~ $ cd log330

~/log330 $ echo "Lorem ipsum" > nouveau_fichier.txt

~/log330 $ echo "Projet réalisé pour L'ÉTS" >> README.md

~/log330 $ git add nouveau_fichier.txt

~/log330 $ git add README.md

~/log330 $ git commit -m "Voici mon premier commit où j'ai ajouté un nouveau
fichier et modifié un fichier existant."

~/log330 $ git push
```

### 3.2.2. Outil de documentation

L'outil de documentation collaboratif de Github est utilisé afin de centraliser la documentation du logiciel et est disponible à l'adresse suivante : <https://github.com/authchir/log330-washing-machine-system/wiki>

### 3.2.3. Outil de gestion des demandes de modifications

L'outil de gestion des demandes de modifications de Github est utilisé afin d'effectuer le suivi des modifications à apporter au logiciel et est disponible à l'adresse suivante: <https://github.com/authchir/log330-washing-machine-system/issues>

Les demandes de modification sont classées dans quatre grandes catégories :

- Amélioration proactive - Perfective
- Amélioration réactive - Adaptative
- Correction proactive - Préventive
- Correction réactive - Corrective

---

Ces quatre catégories permettent de représenter les différents contextes dans lesquels peuvent se présenter une demande de modification: amélioration vs. correction et proactive vs. réactive. Une amélioration correspond à la modification du logiciel afin de lui apporter une valeur ajoutée alors qu'une correction vise à supprimer un défaut déjà présent dans le logiciel. De même, une modification proactive est réalisée avant qu'un besoin ou défaut ne se manifeste, alors qu'une modification réactive est réalisée suite à l'apparition d'un besoin ou défaut.

À titre d'exemple, une demande de modification concernant une défaillance de la conversion des degrés Fahrenheit en degrés Celsius appartient à la catégorie des corrections réactives et la modification effectuée sera donc corrective.

À l'opposé, une demande de modification concernant l'ajout d'un cycle de lavage pour les vêtements délicats appartient à la catégorie des améliorations proactives et la modification posée sera donc perfective.

Lorsqu'un groupe cohérent de modifications est identifié et classifié, une étape jalon est alors créée où la résolution de ce groupe de modifications devient alors le critère définissant l'accomplissement du livrable correspondant.

### 3.3. Directives de livraison

Les directives de livraison sont présentées à l'annexe A de ce plan.

---

## 4. Ressources matérielles, calendrier et budget

### 4.1. Ressources matérielles

- Quatre ordinateurs avec les logiciels requis pour le développement logiciel.
- Un contrôleur du même type qui sera utilisée pour installer le logiciel embarqué de la machine à laver.
- Une carte ayant des ports permettant la connexion avec le contrôleur.

### 4.2. Calendrier des tâches du projet

**Rédaction du modèle d'affaire et du code de déontologie** : Débute le 13 septembre 2013 et se termine le 20 septembre 2013. Nécessaire à la bonne planification du projet.

**Rédaction du document de planification du projet et installation de l'environnement de travail** : Débute le 20 septembre 2013 et se termine le 4 octobre 2013. Nécessaire pour bien comprendre les exigences du projet et également pour comprendre l'ampleur du projet afin de pouvoir analyser et définir ultérieurement les exigences.

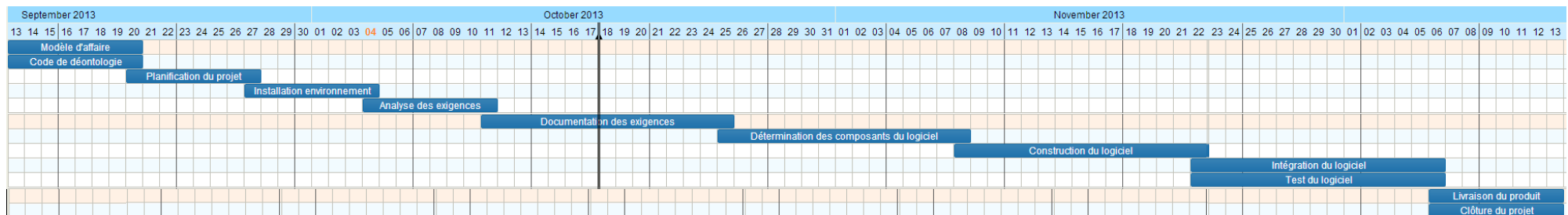
**Analyse et documentation des exigences** : Débute le 4 octobre 2013 et se termine le 25 octobre 2013. Nécessaire afin de bien comprendre les exigences du logiciel pour pouvoir, par la suite, déterminer les composants logiciels cruciaux au bon déroulement du projet.

**Détermination des composants du logiciel** : Débute le 25 octobre 2013 et se termine le 8 novembre 2013. Nécessaire afin de bien comprendre quels composants sont utilisés pour la création du logiciel et pour pouvoir choisir les bons composants selon les critères établis. Cette étape est obligatoire pour débiter la construction du logiciel de manière efficace.

**Construction du logiciel** : Débute le 8 novembre 2013 et se termine le 22 novembre 2013. Partie cruciale du cycle de projet qui permettra de livrer le produit suite à sa conception respectant les contraintes établies dans les documents précédents. Il ne peut y avoir les étapes suivantes, soit l'intégration et le test du logiciel sans la construction de celui-ci.

**Intégration et test du logiciel** : Débute le 23 novembre 2013 et se termine le 6 décembre 2013. Permet de valider le logiciel et de trouver les défauts de dernière minute afin de s'assurer que le tout est conforme aux exigences et que le logiciel opérera de manière acceptable une fois en production. Dernière étape avant la livraison du produit et la clôture du projet.

**Livraison du produit et clôture du projet :** Étapes qui servent à donner au client ce pour quoi il a embauché notre équipe. Il faut aller l'installer en production sur ses postes ou, dans notre cas, sur les contrôleurs embarqués pour les machines à laver. Ensuite, nous pouvons conclure le projet puisque tout le devis a été accompli.



### 4.3. Budget

Non applicable. Nous possédons déjà tout le matériel nécessaire au développement du logiciel.

### 5. Gestion des risques

La colonne « P » est la probabilité d'occurrence du risque, notée entre 1 et 5 (de très peu probable à garanti de se produire).

La colonne « C » indique la conséquence si le risque se transforme en problème, exprimée entre 1 et 5 (de conséquence minimale à conséquence catastrophique).

La colonne « E » indique l'exposition au risque et représente une valeur relative de la priorité. L'exposition est calculée en multipliant les valeurs en « P » et en « C ».

Numéro d'identification du risque	Description du risque	P	C	E	Réduction du risque
1	Défaillance du logiciel	2	4	8	- Documenter le système. - Faire les tests.
2	Dépasser le budget convenu.	3	2	6	- Étude du marché. - Prendre de bons fournisseurs.
3	Ne pas livrer dans les délais alloués.	1	4	4	- Faire des réunions régulièrement pour s'assurer de l'avancement du projet.

Tableau 1. Grille de documentation des risques