

Road Runner

Plan de projet

Cuiseur de riz

Version 1.3

**Israël Hallé
René-Alexandre Giroux
Mathieu Guérin
Martin Grenier**

07-03-2013

Historique des révisions

Date (jj-mm-aaaa)	Version	Description	Auteur
03-02-2013	1.0	Ajout des sections #1 et #2 du plan de projet	Israel Hallé René-Alexandre Giroux Mathieu Guérin Martin Grenier
11-02-2013	1.1	Complétion du plan de projet	Israel Hallé René-Alexandre Giroux Mathieu Guérin Martin Grenier
15-02-2013	1.2	Révision	Israël Hallé
07-03-2013	1.3	Correction après l'audit	Israël Hallé

Définitions

Terme	Définition
Système embarqué	Système informatique qui est intégré à un dispositif, à une machine ou à un autre système et qui pilote ce dispositif, cette machine ou ce système.
ISO/IEC 29110	Norme concernant l'ingénierie de systèmes, l'Ingénierie du logiciel et leur processus de cycle de vie pour les très petits organismes.

Abréviations/acronymes

Abré./Acro.	Définition
ACME	Organisation à l'initiative du projet de cuiseur de riz. Client de Road Runner.
IDE	Environnement de développement intégré

Table des matières

1.Introduction.....	5
1.1 Vue d'ensemble du projet.....	5
1.2 Références.....	5
1.2.1 Références.....	5
1.2.2 Références normatives.....	5
1.3 Profil de libertés/contraintes.....	6
1.4 Identification de la criticité du projet.....	6
1.5 Portée.....	7
1.6 Livrables.....	8
Livrable sortant de gestion de projet.....	8
Livrable interne de gestion de projet.....	8
Livrable sortant de la mise en œuvre logiciel.....	8
Livrable interne de la mise en œuvre logiciel.....	8
2 Organisation du projet.....	9
2.1 Composition de l'équipe du projet.....	9
2.2 Stratégie de contrôle des versions.....	9
2.3 Directives de livraison.....	10
3 Ressource matérielle, calendrier et budget.....	11
3.1 Ressources matérielles.....	11
3.2 Calendrier des tâches du projet.....	11
3.3 Budget.....	11
4 Gestion des risques.....	12
Annexe A – Directives de livraison.....	15

1. Introduction

1.1 Vue d'ensemble du projet

Ce document planifie le projet logiciel du cuiseur de riz d'ACME. Le cuiseur de riz permet à un utilisateur de cuire le riz en prenant soin de cuire avec la bonne température et la bonne durée. Une équipe externe s'occupe du développement du matériel électronique alors que notre équipe s'occupe uniquement de la partie logicielle embarquée.

Nos objectifs sont donc de développer un nouveau modèle de cuiseur de riz comportant 3 modes de cuisson selon le profil basique de la norme ISO/IEC 29110.

1.2 Références

1.2.1 Références

Énoncé des travaux de la société ACME, 2013-01-28

https://cours.etsmtl.ca/log330/private/Travaux/TP_2_Plan_du_projet/Acme%20Enonce%20des%20travaux.pdf

Road Runner - GABARIT - Plan de projet, 2013-01-28

https://cours.etsmtl.ca/log330/private/Travaux/TP_2_Plan_du_projet/Road%20Runner%20GABARIT%20Plan%20de%20projet.docx

Road-Runner - GABARIT - Formulaire de directives de livraison, 2013-01-28

https://cours.etsmtl.ca/log330/private/Travaux/TP_2_Plan_du_projet/Road%20Runner%20GABARIT%20Directives%20de%20livraison%20du%20logiciel.xlsx

1.2.2 Références normatives

ISO/CEI TR 29110-5-1-2, 2011-05-15, Première édition

https://cours.etsmtl.ca/log330/private/Travaux/TP_2_Plan_du_projet/ISO_29110-5-1-2_2011%208F%29-1.pdf

1.3 Profil de libertés/contraintes

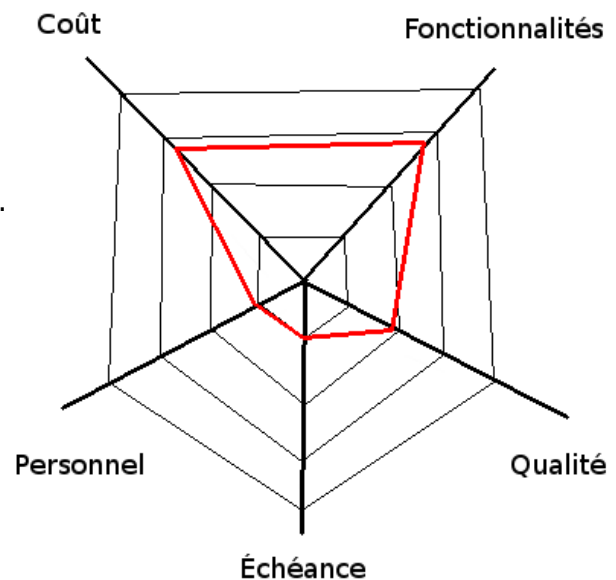
Dimension	Objectif	Contrainte	Degré de liberté
Coûts		Sous contrat	
Fonctionnalités	La version finale du logiciel devra contenir son ensemble de fonctionnalités		
Qualité		Maintenance du produit final très coûteuse	
Échéance			Le produit final doit être remis le 11 avril 2013
Personnel		Quatre personnes	

Aucune marge n'est prévue pour le coût. La liberté est nulle puisque nous sommes à contrat.

Le logiciel doit être complété à la fin du projet, il doit être fonctionnel et de qualité. Il ne peut pas contenir de défaut majeur vu l'impossibilité de mettre à jour le logiciel des cuiseurs de riz vendu sans faire de rappel.

L'échéance n'est pas modifiable. Aucun retard n'est toléré par le client.

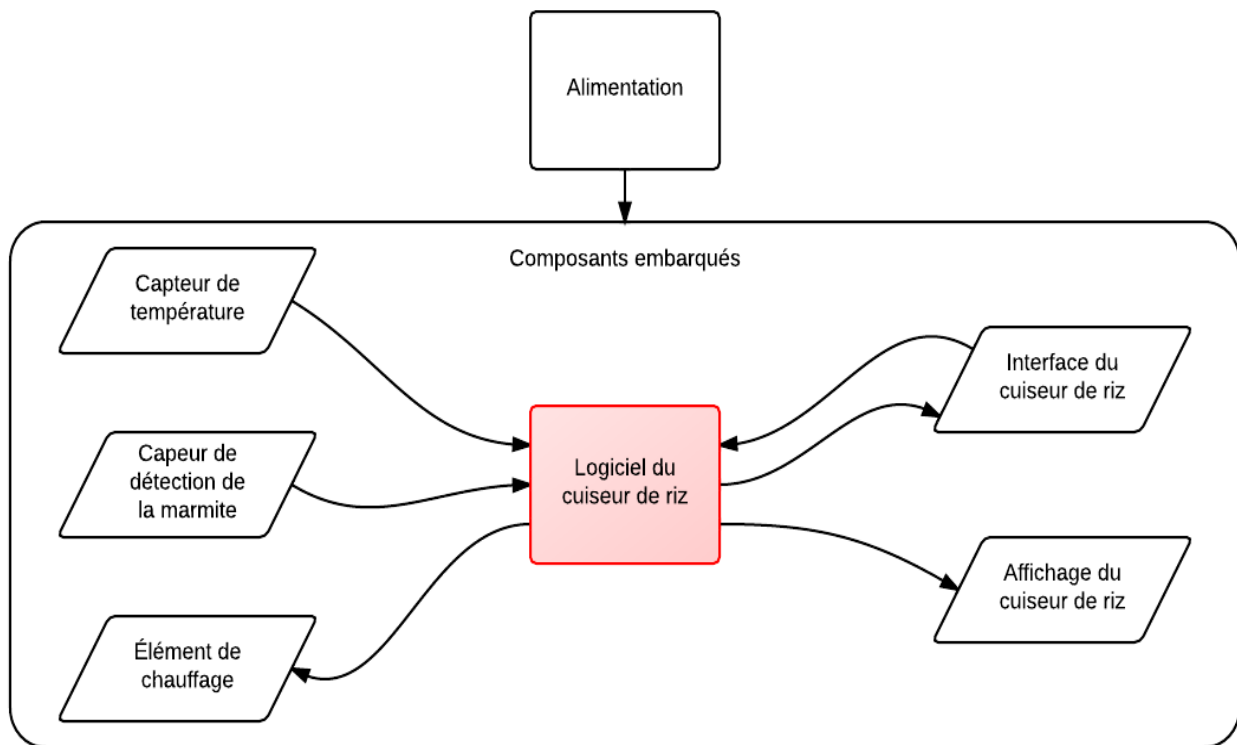
Le personnel est fixé à 4 personnes et ne peut pas changer.



1.4 Identification de la criticité du projet

Dans le cadre de notre projet, il y a un enjeu très important, qui est la sécurité des consommateurs de notre produit. En effet, nous ne voulons pas que notre cuiseur de riz ait une défaillance en pleine utilisation qui pourrait entraîner une surchauffe de l'élément chauffant et donc prendre feu. Il va donc falloir être très vigilant et rigoureux lors des tests, pour détecter toutes les anomalies possibles pour ainsi nous assurer de la sécurité de nos consommateurs. De plus, il faut nous assurer de bien calculer et vérifier la température à laquelle le cuiseur de riz va chauffer, car, s'il est trop chaud, nous allons brûler le riz et s'il est trop froid, le riz ne cuira pas.

1.5 Portée



Notre projet est limité au composant logiciel. Une équipe externe s'occupe de concevoir et développer les composantes électroniques du cuiseur de riz. Ces composantes fournissent les entrées et sorties montrées dans le graphique ci-haut que notre logiciel utilisera afin d'interagir avec l'utilisateur et de contrôler la température du cuiseur de riz.

1.6 Livrables

Voici la liste des livrables du projet de cuiseur de riz produit par l'organisation Road Runner. Tous les livrables sont entreposés dans le répertoire Git du projet.

Livrable sortant de gestion de projet

Livrables	Responsable
Plan de projet	René-Alexandre Giroux
Procès-verbal de la réception	René-Alexandre Giroux
Dépôt d'information du projet	René-Alexandre Giroux
Compte rendu de réunion	René-Alexandre Giroux
Configuration du logiciel	René-Alexandre Giroux

Livrable sortant de la mise en œuvre logiciel

Livrables	Responsable
Configuration du logiciel	Israel Hallé
Spécification des exigences	Martin Grenier
Conception du logiciel	Mathieu Guérin
Enregistrement de la traçabilité	Martin Grenier
Composants logiciels	Israel Hallé
Logiciel	Israel Hallé
Cas et procédures de test	Mathieu Guérin
Rapport des tests	Mathieu Guérin
Guide d'opération du produit	Martin Grenier
Documentation de l'utilisateur du logiciel	Martin Grenier
Documentation de la maintenance	Israel Hallé
Demandes de changement	René-Alexandre Giroux

2 Organisation du projet

2.1 Composition de l'équipe du projet

Rôles	Membre de l'équipe
Analyste	Martin Grenier
Concepteur	Mathieu Guérin
Programmeur	Martin Grenier Israel Hallé
Gestionnaire de projet	René-Alexandre Giroux
Chef Technique	Israel Hallé
Équipe de travail	Martin Grenier Mathieu Guérin René-Alexandre Giroux Israël Hallé

2.2 Stratégie de contrôle des versions

Un projet privé hébergé par Github sera utilisé pour le suivi des billets, la mise en place d'un wiki ainsi que la gestion de version du logiciel. Les différentes tâches de la mise en oeuvre seront créées sur Github afin de les assigner aux différents développeurs. La documentation technique pourra être disponible sur le wiki du projet. Finalement, un répertoire Git nous permettra de gérer efficacement les versions de notre projet. Tous les livrables externes et internes seront disponibles dans le répertoire Git.

Le projet Github est disponible à cette adresse:

<https://github.com/isra17/log330-tp1>. Notez que le projet est privé et seul les membres du projet y ont accès. Ainsi, les développeurs auront besoin d'un compte Github afin de travailler sur le projet. Le logiciel Git est nécessaire afin de récupérer les sources. Plusieurs versions sont disponibles selon la plateforme.

Google Doc est aussi utilisé pour gérer les documents administratifs tel que ce plan de projet. Ceci nous permet de travailler collaborativement sur un document et de gérer ses différentes versions.

Les sauvegardes sont gérées automatiquement par les services utilisés.

La livraison se fera par un courriel à l'aide d'un compte Gmail. Les livrables seront hébergés sur GitHub et le client pourra récupérer une version à partir d'archive directement sur le site du projet.

Le projet sera développé en C à l'aide de l'IDE Eclipse CDT.

2.3 Directives de livraison

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

3 Ressource matérielle, calendrier et budget

3.1 Ressources matérielles

Le projet a à sa disposition 4 postes de travail avec Windows 7 et Ubuntu 12.04. Chaque membre de l'équipe de travail a à sa disposition une licence de Visual Studio et peut utiliser Eclipse CDT sur Linux. Le logiciel TortoiseGit sur Windows afin de cloner le répertoire sur Github. L'outil Git est utilisé en ligne de commande sur Linux. Finalement, LibreOffice est utilisé pour les besoins de traitement de texte.

3.2 Calendrier des tâches du projet

ID	Nom de tâche	Terminer	Début	Durée	janv. 2013			févr. 2013			mars 2013				avr. 2013			
					13-1	20-1	27-1	3-2	10-2	17-2	24-2	3-3	10-3	17-3	24-3	31-3	7-4	
1	Planification du projet	2013-02-04	2013-01-14	4j														
2	Exécution du plan de projet	2013-03-25	2013-02-04	8j														
3	Évaluation et contrôle du projet	2013-03-18	2013-01-14	10j														
4	Clôture du projet	2013-04-08	2013-01-28	11j														
5	Initiation de la mise en oeuvre	2013-02-18	2013-01-28	4j														
6	Analyse des exigences	2013-03-18	2013-01-14	10j														
7	Architecture et conception	2013-03-18	2013-02-11	6j														
8	Construction du logiciel	2013-03-11	2013-01-14	9j														
9	Intégration et tests	2013-03-25	2013-01-14	11j														
10	Livraison du produit	2013-04-08	2013-01-28	11j														

3.3 Budget

Le projet devrait durer environ 84 jours avec quatre ressources. Les ressources travaillent avec un taux horaire de 30.50\$/h et font 2 heures par jour. Donc, on estime à environ 20 500\$ le projet pour le développement du logiciel du cuiseur de riz ainsi que ses livrables.

4 Gestion des risques

La colonne 'P', est la probabilité d'occurrence du risque, exprimée par un nombre entre 1 (très peu probable) et 5 (garantie d'arriver).

La colonne 'C', indique la conséquence si le risque se transforme en problème, exprimé par un nombre compris entre 1 (conséquence minimale) et 5 (conséquence catastrophique).

La colonne 'E', indique l'exposition au risque.

Tableau 1. Grille de documentation des risques

Numéro d'identification du risque	Description du risque	P	C	E	Réduction du risque
1. Criticité	Une défaillance du logiciel peut causer des brûlures ou des incendies. Une mauvaise gestion de la température peut aussi causer la perte du riz.	3	5	15	Mettre plus d'effort lors du développement et des tests pour s'assurer que le cuiseur de riz ne dépasse jamais la température critique (140C)
2. L'incertitude des besoins et exigences des utilisateurs	L'énoncé des travaux est très sommaire et le logiciel dépend du matériel, des entrées/sorties et autres exigences qui peuvent changer au cours du projet.	4	4	16	Clarifier les exigences avec le client, s'assurer qu'elles sont bien comprises par tous les partis et utiliser un modèle itératif pour mieux gérer les changements possibles.
3. Gamme d'environnements	Étant donné que le cuiseur de riz peut être vendu en Asie ou en Amérique, le logiciel doit pouvoir opérer des appareil avec des voltages différents et dans des températures et pression	3	4	12	S'informer si le matériel fonctionne de la même manière, peu importe le voltage utilisé, ou si le logiciel doit opérer différemment. Prendre en compte que

	différente.				l'environnement peut changer la vitesse et la température d'ébullition.
4. Coût de correction des erreurs	Étant donné que le logiciel est embarqué, il n'est pas possible de simplement envoyer les corrections aux clients. Il serait donc nécessaire de faire un rappel extrêmement coûteux.	2	5	10	S'assurer que le logiciel répond à toutes les exigences lors de la livraison puisque les corrections ne peuvent être appliquées aux produits vendus. Mettre de l'accent sur l'assurance qualité.
5. Réglementation	Le client demande d'utiliser le profil basique de la norme ISO/IEC 29110. Il faut aussi s'assurer de respecter les normes qui existent pour réglementer les produits électroniques domestiques.	2	3	6	S'assurer de la compréhension et du suivi de la norme ISO/IEC 29110 par l'équipe logiciel. Faire appel à un juriste pour être aux courants des normes sur les objets électroniques et qui pourraient ajouter des contraintes au développement du logiciel.
6. Taille du projet	Le projet est relativement petit, mais requiert beaucoup de temps mis sur la qualité du logiciel et sa fiabilité étant donné la criticité et le coût des corrections des erreurs.	2	4	8	Avoir une bonne planification qui s'assure d'avoir le temps requis pour bien valider le logiciel.
7. Communication	Le développement du logiciel dépend beaucoup du matériel embarqué développé en parallèle par une autre organisation. Une	3	3	9	S'assurer d'une bonne communication entre l'équipe de développement du logiciel et celle du

	mauvaise communication des changements ou des exigences entre les deux organisations pourrait causer des problèmes au logiciel.				matériel pour partager tout changement et exigence qui pourrait avoir un impact sur le logiciel.
8. Culture de l'organisation	Les développeurs des petites entreprises ont l'habitude de s'épanouir en conservant et développant de nouveaux projets tout en négligeant l'assurance qualité. Le projet requiert beaucoup d'énergie sur l'assurance qualité.	2	5	10	La norme ISO/IEC 29110 devrait fournir les outils nécessaires à implanter des processus efficaces et adaptés au besoin du projet.

Annexe A - Directives de livraison