Etat : En cours
V2.0

AppInfo

Votre logiciel en toute sérénité

29/09/2014

PLAN D'ASSURANCE QUALITE

Projet : 24/24Manager

Client : CPE Lyon

LYON CPE
ÉCOLE SUPÉRIEURE DE CHIMIE PHYSIQUE ÉLECTRONIQUE DE LYON

Julien BRIOT – Fabien CHAMPEL – Justine POYARD
Long LE DAC – Kim HERR
GROUPE G

Objet :

Ce Plan d'Assurance Qualité (PAQ) expose les différentes règles et processus qui seront appliqués durant le développement du projet 24/24Manager par l'entreprise AppInfo. L'objectif est d'assurer la qualité du projet 24/24Manager tout le long de son développement jusqu'à la réalisation finale.

Acteurs du projet :

Maitrise d'ouvrage : CPE Lyon

Maitrise d'œuvre : AppInfo

Table des matières

I. Introduction

1. Descriptif du projet

Dans le cadre de la formation d'apprenti ingénieur en Informatique et Réseaux de communication, il nous a été demandé de réaliser un projet complet (du dossier de lancement à la livraison) au sein du module génie logiciel. Ce projet est réalisé par groupes de cinq personnes.

Lors de la réalisation du dossier de lancement, il nous a fallu choisir entre deux sujets :

- Logiciel de gestion d'approvisionnement d'un magasin 24/24
- un jeu « Attention Pirates »

Après l'analyse des compétences de notre équipe, nous nous sommes rendu compte que le sujet le plus adapté était le premier sujet.

2. Objectif du plan qualité

L'objectif de ce PAQ est de définir en accord avec le client, les moyens mis en œuvre pour obtenir la qualité nécessaire à la bonne réalisation du projet.
Il a pour but de détailler :

- L'organisation du projet
- Les documents applicables dans le cadre du contrat
- Les documents à rendre
- Les différentes terminologies
- La méthodologie applicable pour le projet
- Le processus de la gestion de la qualité
- Les différentes normes applicables

3. Champs d'application du PAQ

Ce document est applicable dès le début de projet et jusqu'à la fin de celui-ci, de ce fait il est consultable uniquement par les membres concernés par ce projet, y compris par CPE Lyon.

4. Objectifs qualité

- Tenue des délais lors de rendus de documents ou code source
- Maintenir des échanges réguliers avec le client / les enseignants
- Répondre au mieux aux besoins utilisateurs
- Respecter les règles établies dans le présent document

5. Terminologie et abréviations

CdP : Chef de Projet

MCT : Modèle Conceptuel de Traitement

MOA : Maîtrise d'ouvrage

MOE : Maîtrise d'œuvre

PAQP : Plan d'Assurance Qualité Projet

PAQL : Plan d'Assurance Qualité Logiciel

RQ : Responsable Qualité

RDC : Responsable(s) documentation et communication

II. Organisation MOA et MOE

1. Organisation client / Relations avec le client

a) Communication entre le client et la MOE

Le client est ici représenté par les enseignants de l'école d'enseignement supérieur de CPE Lyon. Les échanges se feront majoritairement par mail, et si possible lors de « réunion » organisées occasionnellement. Le client aura accès à tous les documents produits durant le projet.

b) Rôle des différents acteurs du client

Les rôles des représentants CPE Lyon sont :

- **Régis Mathieu :** responsable qualité
  Il est responsable de l'étude et de la validation du Plan d'Assurance Qualité rédigé par AppInfo.

- **Xavier Trouillot :** responsable du besoin et contact prioritaire
  Il a pour rôle de répondre aux demandes d'AppInfo concernant les besoins de CPE Lyon sur le projet. Il participe également à la validation du Plan Assurance Qualité, de la conception proposée et de la réalisation finale.

- **Nicolas Padey :** responsable analyse et conception
  Il est responsable de l'étude et de la validation de la conception proposée par AppInfo.

- **Jérome Thevenon :** responsable spécifications et réalisation finale
  Il est responsable de l'étude et de la validation des spécifications proposées par AppInfo. Il vérifie aussi la conformité de la réalisation finale par rapport à la demande initiale.

2. Organisation MOE

a) Membres et rôle des différents acteurs d'AppInfo

Chef de projet : Julien BRIOT

Rôle:

- Mise en place de l'équipe de projet et distribution des rôles
- Constitution du planning prévisionnel pour les tâches à effectuer
- Coordination de l'équipe du projet

- Vérification du bon déroulement du projet
- Validation et présentation des résultats aux clients

Responsables communication et documentation : Kim HERR & Justine POYARD

Rôle:

- Rassembler les informations collectées par chacun
- Veiller à ce que les livrables intermédiaires soient faits et dans les temps
- Communication avec la MOA (les enseignants) par mail / en cours
- Diffusion des documents dans le groupe de projet, les MOE,
- Dialogue au sein du groupe

Responsable qualité : Fabien CHAMPEL

Rôle:

- Veiller à la bonne mise en place d'une démarche qualité
- Veiller à la qualité de tous les livrables intermédiaires
- Veiller à la qualité du livrable final pour le client

Responsable technique : Long LE DAC

Rôle:

- Rechercher les outils les plus adaptés à chaque problème.
- Vérification de la bonne utilisation des outils
- Vérification des choix techniques (faisabilité)

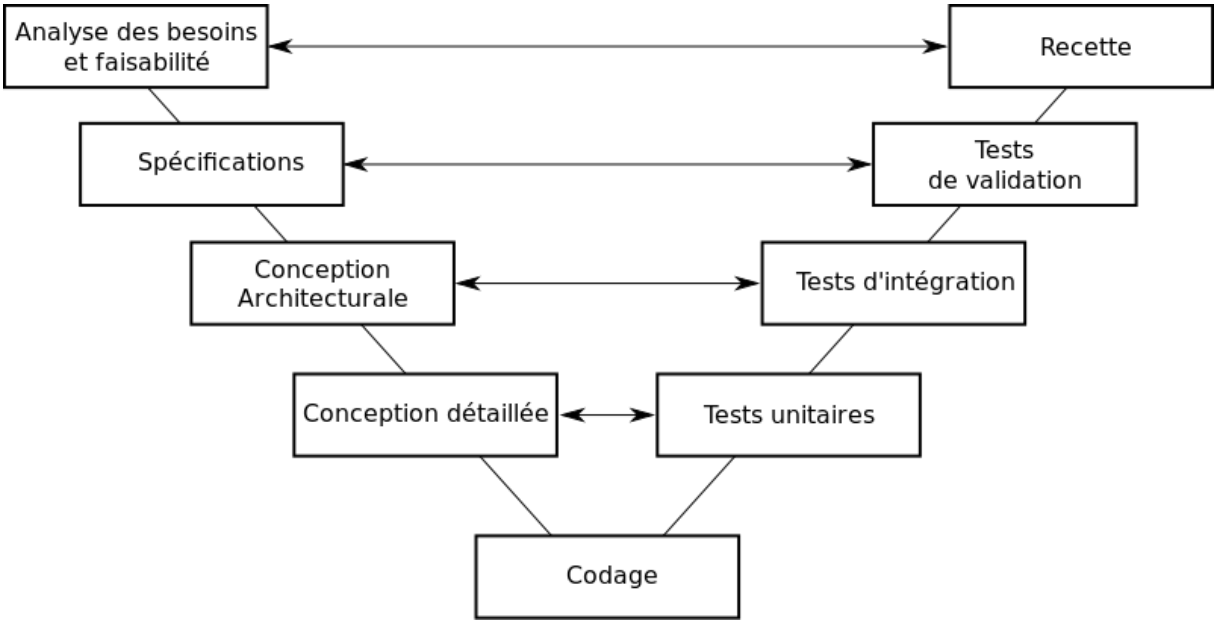
b) Matrice de compétences

Compétences ou Domaines	Niveau Requis	Julien Briot	Fabien Champel	Kim Herr	Long Le Dac	Justine Poyard
Qualité	1	1	1	1	1	1
Communication	3	2	2	3	2	2
UML	3	2	2-3	1	3	2
JAVA	3	3	2	2	3	2
Ergonomie	3	2	1	3	2	1
SQL	3	3	3	1	3	3

Légende : 1 – Connaissances Théoriques
2 – Connaissances Pratiques
3 – Bonnes Connaissances
4 - Expert

III. Qualité au niveau processus de développement

1. Processus du projet



Processus Cycle en V

La réalisation de l'application se déroulera en 3 grandes parties (chacune se découpant elle-même en sous-parties):

- Définition du besoin - conception
- Réalisation - codage
- Validation – tests

Nous avons choisi le « Cycle en V » pour la clarté qu'il impose dans le rôle de chacune des personnes de l'équipe.

La validation de chaque étape nécessite l'adhésion de tous les membres de l'équipe, ce qui diminue le risque de conflit dans la suite du projet.

Nous nous engageons donc à suivre les étapes décrites par ce processus, pour le bon déroulement de ce projet de Génie Logiciel.

2. Phases de projet

a) Définition du besoin – conception

➢ *Analyse du cahier des charges*

Lors de cette première étape, nous avons pris connaissance de la demande du client. Dans notre cas, il nous est demandé de réaliser un logiciel de gestion d'approvisionnement en pain, viennoiseries et boissons d'un magasin. Il est important d'anticiper les différentes technologies qui pourraient être utilisées pour le réaliser. Un planning sera déduit de cette phase d'étude.

➢ *Spécifications*

Cette étape permettra de définir l'ensemble des fonctionnalités qui seront implémentées dans le programme et de définir la portée du projet.

➢ *Conception*

La conception définira la structure du programme ainsi que l'organisation des différents éléments du système. Cette partie sera réalisée lors de l'analyse UML avec la création de diagrammes de séquences ainsi que de diagrammes de classes qui seront ensuite implémentés lors de l'étape de développement.

Il faudra par la suite détailler précisément la conception globale par module afin de pouvoir les développer et préparer les tests unitaires. Elle comprend les diagrammes d'activité et les diagrammes état-transition. Le document de conception doit être validé par l'équipe projet.

b) Réalisation – codage

Lors de cette phase, les différentes classes et méthodes seront implémentées afin de mener à bien ce projet tout en suivant scrupuleusement les analyses précédemment effectuées.

c) Validation – Test

La phase de test intervient ensuite pour **identifier les bugs** potentiels et vérifier le bon respect des cahiers charges. Il y aura tout d'abord des tests unitaires, pour vérifier que toutes les fonctionnalités décrites dans le dossier de spécification se déroulent sans erreur, puis un test de validation générale du programme.

Dans le cadre de notre projet, la phase de suivi et de pilotage de projet n'aura pas lieu mais lors de projets en conditions réelles, elle est indispensable pour la pérennité du logiciel.

IV. Documentation

1. Documents références

Les clients ont fourni à notre groupe plusieurs documents auxquels se référer lors de ce projet :

a) Planning Prévisionnel

Un planning prévisionnel du développement du projet sera établi à l'aide d'un diagramme de GANTT et présenté dans un document ultérieur. Chaque fonctionnalité aura un nombre de jours attribué, un auteur et une date limite de rendu.

Voici le planning prévisionnel de rendu des livrable au client :

- Plan d'Assurance Qualité : 18 septembre 2014
- Dossier de spécification : 23 octobre 2014
- Dossier de conception : 11 novembre 2014
- Démonstration du logiciel et rendu du code source : 5 janvier 2015

Ces rendus se feront via le site web e-campus de CPE Lyon, dans les dossiers indiqués par les clients.

b) Explicitation du sujet

Un document nous a été fourni par le client afin d'expliciter brièvement le fonctionnement du magasin de vente. Il contient une description des fonctionnalités demandées par le client.

Par exemple :

« En arrière-boutique, on cuit les pains et les viennoiseries, qui sont préalablement stockées crues sous forme congelée, lorsque le nombre de produits correspondant disponible passe en dessous d'un certain seuil (définissable par produit et variable suivant l'heure).

La quantité de produit à cuire est également paramétrable en fonction de l'heure et de la fréquentation de l'aéroport.

Lorsque l'opérateur de cuisson prend en charge une cuisson, il signale la prise en charge en début de préparation et la disponibilité en fin de cuisson. »

2. Documents applicables

AppInfo met à disposition trois types de documents à compléter afin de rendre plus simple les relations entre la MOE / MOA :

- Demande d'information : fiche pour toute question de la MOE envers la MOA ou inversement.
- Demande de Modification de la documentation : plusieurs demandes sont faites sur une même fiche.
- Demande de Modification logicielle : une demande unique par fiche avec suivi de l'avancement de la résolution.

3. Liste des documents de gestion projet

Pour assurer un bon suivi de projet, les documents suivant sont nécessaires :

a) Dossier de lancement de projet

- Répartition des rôles dans ce projet
- Etablissement du planning prévisionnel
- Maîtrise des compétences des développeurs
- Suivi des risques

b) Suivi de tâche

- Détail de l'avancement des tâches selon les développeurs pour suivre l'évolution des tâches.
- Permettra de percevoir les difficultés s'il y en a et de les résoudre plus facilement.

c) Dossier bilan

- Evaluer l'organisation du projet
- Analyser les objectifs techniques et les résultats, déterminer si des améliorations sont possibles.
- Bilan qualité, estimation des coûts qualité et des coûts de non qualité (résolution des bugs)

4. Liste des documents techniques et de réalisation

Les documents listés ci-dessous servent pour la partie technique et la réalisation du projet afin de définir en détail le projet :

- Dossier de spécification : document précisant les fonctionnalités détaillées du produit, les interactivités avec les utilisateurs. Il renseigne les tâches à effectuer et les composants à développer.
- Dossier de conception : détails des modèles conceptuels (données, traitement), architecture du logiciel et diagramme UML.
- Plan de test de validation : description générale de la structuration et de l'organisation des tests.
- Dossier de test de validation : contient la description de tous les tests à effectuer sur l'application
- Manuel d'utilisation : document permettant de décrire le fonctionnement du logiciel fourni au client, afin qu'il puisse utiliser au mieux l'application.

V. Méthodes, Outils et Règles

1. Méthodes et méthodologies retenues

a) Merise

Il s'agit d'une méthode de conception, de développement et de réalisation de projet informatique. Celle-ci se base sur la séparation des données et des traitements à effectuer en plusieurs modèles : conceptuel, de traitement, physique.
La séparation des données et des traitements permet de cibler les parties à retoucher sans pour autant modifier le reste. En effet, l'agencement des données n'a pas à être souvent remanié alors que les traitements le sont fréquemment.

Afin de décrire un traitement ou un ensemble de traitements, on utilisera un MCT (Modèle Conceptuel de Traitement) de Merise.
Pour toute description de données et de relations entre les données on utilisera le MCD de Merise (Modèle Conceptuel de Données).

b) Java

Le client demande que le langage de programmation soit Java. C'est un langage de programmation moderne développé par Sun Microsystems (aujourd'hui racheté par Oracle).
Une de ses plus grandes forces est sa portabilité : une fois votre programme créé, il fonctionnera automatiquement sous Windows, Mac, Linux, etc.

On peut faire de nombreuses sortes de programmes avec Java :

- des applications, sous forme de fenêtre ou de console ;
- des applets, qui sont des programmes Java incorporés à des pages web ;
- des applications pour appareils mobiles, avec J2ME ;
- et bien d'autres comme J2EE, JMF, J3D pour la 3D...

Nous coderons sous forme de fenêtre avec l'aide du framework graphique standard Swing inclus dans Java.

2. Outils (Outils logiciels, autres outils)

Pour tous les outils utilisés lors de ce projet, les licences seront valides.

- Word 2010/2013 pour la rédaction des livrables
- Google Doc/Drive pour les communications et le partage en groupe
- Eclipse pour le développement en Java
- GANTTProject pour l'établissement d'un planning prévisionnel
- Quencepterise pour la génération de diagrammes Merise
- ASTAH pour les modélisations UML

- GITHub pour le travail en équipe et le partage des sources. Lors de la rédaction de ce document les configurations ne sont encore pas totalement fonctionnelles.
- Le logiciel SourceTree pour l'interface graphique de partage GIT.

3. Règles et normes de codage

a) Normes de développement en Java

Les classes

Les classes seront nommées selon la méthode camelCase mais devront commencer par une majuscule. Il faudra utiliser des mots qui décrivent bien la classe tout en n'étant pas trop long.

Exemple : MenuPrincipal, ViennoiseriesCuites, etc.

Les interfaces se nommeront de la même manière que les classes.

Les méthodes

Les méthodes seront nommées selon la méthode camelCase et comporteront généralement un verbe d'action. Cela donne par exemple : cuitLaViennoiserie(), calculDeLaRecette(), vendre(), etc.

Les variables

Les variables se nommeront également en camelCase et en commençeront par une lettre (a-z). Le nom d'une variable devra être court et clair. Les variables à 1 caractère seront proscrites sauf pour un usage temporaire (i, j, k, l, m, n pour les entiers et c, d, e pour les caractères). Cela donne par exemple : nombrePainChocolat, recetteFinale, i, etc.

Les constantes

Les constantes devront être écrites en majuscules et pour séparer les mots les "_" sont à utiliser. Cela donne par exemple : LARGEUR_MAX, ID_COURANT, etc.

Les commentaires

Chaque méthode devra faire l'objet de commentaires indiquant son rôle et celui de chaque variable ainsi qu'une description des entrées / sorties / entrées-sorties. Ces commentaires serviront à l'élaboration d'une documentation (JavaDoc)

4. Règles de gestion et de structuration des documents

a) Identification des documents

L'identification d'un document sera indispensable pour la gestion de la documentation du projet. Le nom du fichier lorsque le document sous forme électronique est de la forme suivante :

*GROUPE G_Nature du document_version.**

Exemple : GROUPE G_PAQ_v1.0.pdf

b) Présentation des documents

Tous les documents (sauf les ébauches) devront suivre la structure suivante :

- police de caractère : Calibri Light, taille : 12
- une page de garde avec les éléments suivants :
  - ✓ le titre du document
  - ✓ le nom du groupe et son logo
  - ✓ la date de rendu
  - ✓ le numéro de version
  - ✓ l'état du document
  - ✓ le nom du Groupe ainsi que celui de ses membres

Pour toutes les mises à jour importantes du document, il faudra indiquer :

- le numéro de la version
- la date de dernière mise à jour de cette version
- l'objet de la mise à jour du document

Chaque page sera composée d'un pied de page contenant :

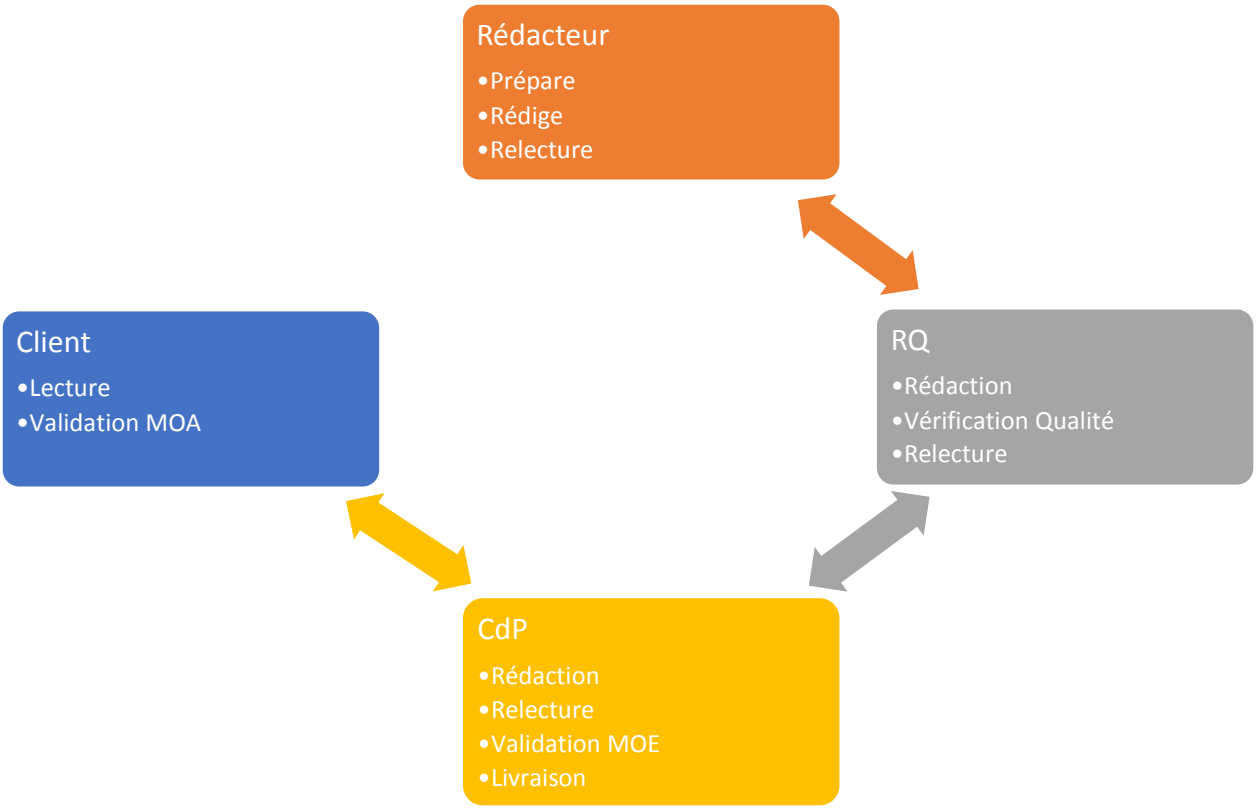
- « Projet Informatique 4IRC »
- « GROUPE G »
- titre du document
- Numéro de page | Nombre de pages total

c) Etat d'un document

Pendant son cycle de vie (document de type livrable uniquement), le document peut se trouver dans les états suivants :

- En cours: le document est en cours d'élaboration
- Finalisé: le document est terminé par l'auteur et prêt à être diffusé
- Vérifié: le document est approuvé par le responsable qualité
- ValidéCP: le document est approuvé par le chef de projet
- ValidéCL: le document est validé par le client

Ci-dessous se trouve le cycle de validation d'un document.



d) Gestion des versions

Les livrables intermédiaires doivent tous disposer d'un numéro de version, il est de la forme suivante : V X.x. Le « X » sera incrémenté pour les modifications majeures, alors que le « x » le sera pour les modifications mineures.

VI. Gestion des modifications

1. Type et origine des modifications

Deux types de modification sont possibles. Les modifications logicielles et les modifications de la documentation.

Les modifications logicielles peuvent avoir plusieurs causes et devront suivre un processus précis pour être validées. Ces modifications sont nécessaires pour assurer la pérennité du logiciel et sont divisées en deux catégories distinctes:

- Une erreur a été détectée et doit être corrigée. Une anomalie sera classifiée selon son niveau de priorité et son impact en termes de ressources à utiliser pour sa résolution.
- Une évolution / mise à jour est nécessaire pour répondre à un nouveau besoin de la Maitrise d'Ouvrage. Comme les erreurs, une évolution sera classifiée sur les deux échelles précédemment citées.

Les demandes de modification de la documentation sont effectuées par le client lorsque celui-ci souhaite apporter une modification à une documentation comme le PAQ par exemple. Ces modifications doivent aussi suivre un processus précis pour être valides. Contrairement aux demandes de modification logicielle, une demande de modification de la documentation n'a aucun niveau de priorité.

2. Mise en œuvre des modifications logicielles

a) Fiche de rapport d'erreur / demande de mise à jour

Notre groupe met à disposition une fiche de type "Template" destinée à la MOA mais aussi à l'équipe de test de la MOE, leur permettant de faire une demande de modification à l'équipe de développement de la MOE. Une fiche par demande devra être éditée et envoyée à l'adresse suivante : groupeg.4irc@gmail.com

Cette fiche permettra de regrouper les informations avec le plus de détails possible, nécessaires au bon déroulement de la modification. Cette fiche sera complétée pendant la résolution de la modification, par la MOE, pour suivre son évolution.

b) Processus de modification

Un processus de modification a été mis en place par le groupe pour permettre la résolution des modifications dans les délais les plus restreints possible. La qualité des modifications respectera les engagements pris dans le Plan d'Assurance Qualité Projet.

- Si la Maitrise d'Ouvrage ou l'équipe test de la Maitrise d'Œuvre détecte une anomalie pouvant entrainer une défaillance perceptible au niveau fonctionnel du logiciel ou souhaite ajouter une nouvelle fonctionnalité au logiciel, une fiche de rapport d'erreur ou demande d'évolution doit être éditée. Cette fiche doit être remplie avec des informations les plus précises possibles. L'utilisation de capture d'écran, listings des actions avant l'apparition d'une anomalie, etc. est vivement conseillée pour accélérer le processus de résolution. La personne éditant la fiche renseigne un degré de priorité non définitif.
- La fiche est envoyée à l'adresse mail traitant les différentes demandes (groupeg.4irc@gmail.com).

- La fiche de demande de modification logicielle est alors traitée une première fois par notre groupe pour s'assurer de la cohérence des informations fournies, les mettre à jour si besoin avec le client et définir un niveau de priorité fixe avec le client.

- Les modifications sont effectuées en pré-production pour correspondre aux besoins énoncés dans la fiche de demande.

- Une nouvelle version du logiciel est alors transmise au client pour être utilisée en production.

- La fiche de demande est clôturée lorsque le client atteste de la bonne implémentation de la modification.

c) Niveau de priorité

Chaque nouvelle demande comporte un niveau de priorité qui définit la rapidité avec laquelle le problème doit être résolu.

Le niveau de priorité est fortement lié au type de problème qui doit être solutionné ou à la nouvelle fonctionnalité à développer. Par exemple, une anomalie retournée concernant un problème d'affichage mineur aura généralement un niveau de priorité faible.

Lorsque le niveau de priorité est fixé entre le client et notre société, celui-ci n'est pas définitif et peut varier selon l'évolution des besoins du client.

La classification du niveau de priorité est différente selon si la demande traite la correction d'une anomalie ou l'ajout d'une fonctionnalité. La priorité d'une nouvelle fonctionnalité sera définie selon le besoin du client alors que la priorité du traitement d'une anomalie sera établie selon le degré d'impact de l'anomalie sur l'utilisation du logiciel. Les niveaux de priorités pour la résolution d'une anomalie sont définis ci-dessous:

Haut: Anomalie bloquante pour l'utilisateur. Une fonctionnalité ne peut être utilisée sur le logiciel et aucune solution ne peut contourner ce problème. La correction doit être mise en place dans une prochaine mise à jour.

Moyen: Anomalie bloquante pour l'utilisateur mais une solution a été mise en place pour contourner le problème. Sa résolution n'est pas urgente mais souhaitable dans l'intérêt du client.

Faible: Anomalie non bloquante pour l'utilisateur tel que des problèmes d'affichages mais n'empêchant pas l'utilisation du logiciel par le client.

3. Mise en œuvre des modifications de la documentation

a) Fiche de demande de modification de la documentation

Comme pour les modifications logicielles, une fiche « Template » est disponible pour permettre au client de faire une demande mais celles-ci permet de regrouper plusieurs propositions de modifications avec une solution proposée. Cette fiche doit être remplie avec des informations détaillées pour permettre de comprendre et d'appliquer correctement en compte les modifications souhaitées. Cette fiche sera envoyée à l'adresse mail précédemment citée.

b) Processus de modification

Un processus simplifié a été mis en place pour faire une demande de modification de la documentation.

- Le client rempli la fiche de demande de modification de la documentation et renseigne les différentes informations demandées.
- La fiche est envoyée à l'adresse mail traitant les différentes demandes (groupeg.4irc@gmail.com).
- La MOA s'assure de la cohérence de la modification à effectuer et la modification est effectuée.
- Une nouvelle version de la documentation est alors transmise au client.
- La fiche de demande est clôturée lorsque le client valide la modification.

VII. Gestion des Risques

Le développement du logiciel peut-être contraint à certains risques qu'il est nécessaire de prendre compte pour éviter de devoir les traiter. Le tableau ci-dessous résume les principaux problèmes qui pourraient apparaitre en précisant un degré importance, l'impact qu'ils auraient sur le développement du logiciel et la solution à appliquer pour résoudre le problème.

Description du risque	Importance (1-5)	Impact	Action
Le non-respect des délais de livraison	5	Non validation du projet en première session	Planning prévisionnel établit. Vérification à chaque séance de l'avancé des tâches à accomplir
Le non-respect des fonctionnalités décrites dans le dossier de spécification	4	Non adéquation avec le dossier de spécification	Mise en place de tests unitaires et tests d'intégrations
Impossible de réaliser une fonctionnalité	3	Non adéquation avec le dossier de spécification	Revoir sa nécessité avec le client Sous-traité
Mauvaise synchronisation des ressources	2	Retarde le projet	Mise en place d'un SVN pour partager les ressources
Le non fonctionnement des programmes sur les postes	1	Livraison impossible	S'assurer la présence et/ou l'importation des bibliothèques nécessaires

VIII. Reproduction, Protection et livraison

1. Précaution à prendre

Aucune procédure de reproduction particulière n'est prévue.

Pour la protection, aucune mesure particulière n'est à mettre en place car nous allons réaliser des prototypes.

Nous nous contenterons de livrer les prototypes à notre responsable aux échéances fixées dans le planning fourni.

La livraison du mois de janvier comprendra le prototype, les fichiers sources ainsi que tous les documents de référence.

La livraison est effectuée par la société AppInfo. Elle contiendra le code source et les documentations de l'application.

Les documents livrables seront fournis dans un dossier «AppInfo». Les règles de nommage de ces fichiers sont explicitées dans la partie V.4.a.

Ce logiciel sera développé dans l'IDE Eclipse sur nos machines personnelles et sur les machines de CPE occasionnellement.

Lors de la livraison du produit, une procédure complète d'installation sera fournie (dans le manuel utilisateur) afin de permettre à des personnes étrangères à l'équipe de développement d'installer et d'utiliser le produit sans difficulté.

2. Modalité de livraison

Règle de nommage :

La règles de nommage est la même que décrite dans la partie ci-dessus (F - IV).

Document type :

- Compte rendu
- Spécifications fonctionnelles générales
- Documentation
- Cadrage fonctionnel (documentations utilisateurs)

Les Livrables seront envoyés au «client» après :

- validation formelle par le chef de projet des éléments du livrable
- prise de connaissance des livrables par tous les membres de l'équipe
- lorsque les procédures ou les critères d'acceptation des livrables de type techniques prévus dans le planning et les spécifications du projet auront été satisfaits.

La contestation de la validation d'un document devra être soumise par le «Client» par mail pour que le document fasse l'objet d'une modification et soit ensuite soumis à nouveau à la validation.

IX. Suivi de l'application sur le plan qualité

1. Principes

L'application du présent PAQ se doit d'être respectée afin d'obtenir un produit final de qualité. Pour cela, chaque membre rattaché au projet devra se conformer aux différentes instructions/règles citées tout au long du PAQ. Le responsable qualité, a quant à lui le devoir de faire appliquer ces règles, dans le cas contraire il devra intervenir.

2. Interventions du RQ

Le responsable qualité à le devoir de veiller à la bonne exécution du PAQ par toutes les parties prenantes du projet. Il sera chargé de valider et de vérifier la qualité de tous les livrables de celui-ci.

Cette personne doit intervenir lorsqu'un membre de l'équipe projet a besoin de renseignements du point de vue qualité et pour le contrôle lorsqu'un composant/fonction est terminé.

3. Procédure à suivre en cas de non-respect

Le responsable qualité se doit en cas de non-respect de ce PAQ de prendre les dispositions nécessaires pour qu'il soit appliqué par tous les membres du projet. Si besoin, il peut faire une modification de ce document qui devra être validé par le chef de projet et les clients.

X. Conclusion

Ce PAQ met en avant les dispositions que la MOE doit suivre. Ce document permet d'expliquer en détail au client les méthodes utilisées lors du développement du logiciel et les processus mis en place pour résoudre les problèmes qui pourraient être rencontrés pendant la production ou après le déploiement du logiciel. L'objectif recherché est que le niveau de qualité atteint soit élevé sans pour autant être trop contraignant pour la MOE.