****

**PLAN D’ASSURANCE QUALITE**

**Projet – M 2 P CCI – Groupe 1**

**Version 1.0**

**27/02/2015**

1. **Présentation**

Le but de ce document est de spécifier les règles qui doivent être prises en compte pour assurer la qualité du projet.

Dans ce document, nous présenterons les différents acteurs qui interagissent dans le cadre du projet, des opérations et normes liées et la liste des documents à produire.

Le projet se déroule dans le cadre d'études du M2 P CCI.

Ce document est indispensable à lire pour tous les membres du projet pour assurer la qualité du travail. Le PAQ décrit:

* Une référence commune à tous les membres de l'équipe du projet avec l'objectif d'établir une cohérence dans les méthodes de travail.
* Les critères de qualité à respecter en relation à la demande du client
* La procédure à suivre et les outils à utiliser avec les règles correspondant.

1. **Responsabilités du projet**

Le suivi du déroulement du plan de qualité et de son application est responsabilité du responsable de qualité.

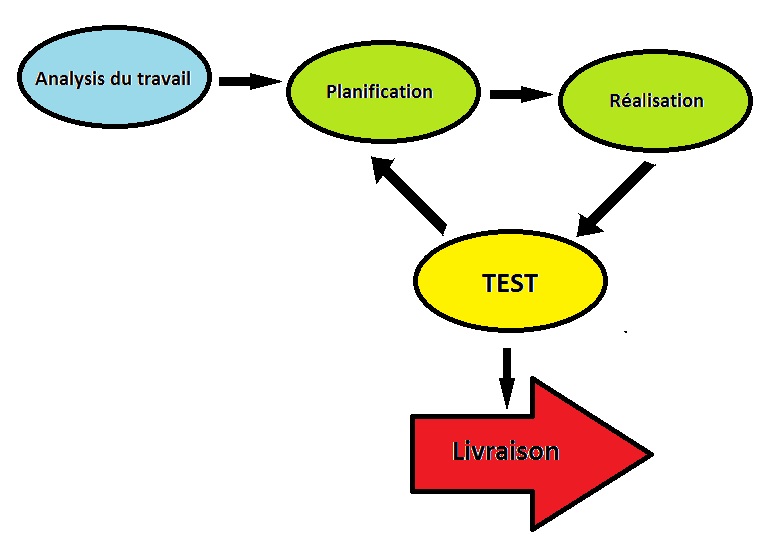
La coordination du déroulement du plan de travail est responsabilité du chef de projet.

L’assignation de responsabilité dans le projet est comme suit :

* Chef / Responsable du projet : Tom BOISSONNET
* Responsable de qualité : Teresa TOYOS
* Gardien du temps : Othmane ZOUHAIR
* Elhadj SAIDOU
* Hassan ABDI

Tous les membres de l’équipe de projet sont autorisés de proposer des modifications pour ce document. Les modifications doivent être valides dans une réunion par décision conjointe du responsable du projet et responsable de qualité.

1. **Cycle de vie du travail**



IMPORTANT: Chaque trois jours, tous les membres d'equipe devrait remplir un formulaire avec l'objetif de creer un bilan retrospectif. Avec ces documents le chef du project peut surveiller le deroulement du projet et prendre les decisions pertinents.

1. **Liste de règles**

*4.1 Taches*

*Nom de tâche* : forme verbale (infinitive)

* **Le nom de tâches doit faire référence aux termes dans le glossaire**

Tache composite : Une tâche composite doit comporter au moins deux sous-tâches

* Une tâche composite est soit abstraite soit du même type que tous ses sous-tâches

Tâche élémentaire : ne peut pas être une tâche abstraite

*Exemples :*

Nom de tâche : *ReserverUnePlace*

* *Place* devrait être dans le glossaire
* Si la signification n’est pas claire il faut faire la définition dans le glossaire aussi. Ici pour exemple, il faut faire la définition de *ReserverUnePlace* si n’est pas claire l’utilisation.

*4.2 Exigence*

Nom d’exigence : doit être clairement une exigence (nom significative)

* Le nom d’exigence doit faire références à des termes du glossaire

Nom d’exigence fonctionnelle : doit commencer pour une forme verbale dans infinitive.

* C’est une fonction à être exécuté par un acteur.

Définition d’exigence : doit être limité à l’expression d’une contrainte imposée sur le système.

* Ne doit pas être une description de :
* Un scenario
* Une suite d’actions
* Une caractéristique relative à l’exigence
* Détailles techniques
* Restrictions

Toute ça on le peut écrire, dans case d’indispensabilité, dans notes associés à l’exigence.

Définition d’exigence fonctionnelle : correspond à une fonctionnalité destinée à un ou plusieurs acteurs.

[SSS] doit permettre à [AAA] de …

[AAA] est le nom du système, ou le nom d l’acteur et définit la fonctionnalité proposé

[SSS] doit permettre à … (il peut être éliminé si c’est évident)

* La première partie fait intervenir le nom du système explicitement, comme ça elle permet de rendre explicite le fait que le system réalise.

Exigence multiple : Le texte qui fait référence à plusieurs exigences simultanément doit être séparé. Ça permit identifier clairement le type de chaque sous-exigence et attribuer à chacune propriétés différents.

* La définition d’une exigence doit être courte et concise.

Priorité d’exigence : doit être clairement exprimé, ils doivent être considérées les unes par rapport aux autres, et doivent être ajoutées au cours d’un projet.

* Pas comporter des formes modales (devrait, on souhaite que,..)
* La définition doit être exprimée de manière impérative

**Attention**

* Exigence incohérente avec une autre exigence décrite
* Exigence Invalide par rapport aux besoins exprimés par le client
* Exigences plus restrictifs que ceux exprimés par le client
* Exigence moins restrictifs par rapport a l’expression de besoins exprimés
* Exigence inadapté aux demandes

*Exemples***:**

Nom d’exigence : *OK*

Nom d’exigence fonctionnelle : *InscrireUneEquipe*

Définition d’exigence : *L’[EquipeTechnique]*

Définition d’exigence fonctionnelle *: [CyberGarage] doit permettre au [ChefDeMagazin] d'enregistrer les [Pieces]*

Priorité d’exigences :

MAL

*DeveloppementJDBC: Il est serait utile que l’interface [JDBC] soit utilisée pour l’accès à la base de données*

BIEN

*L’interface [JDBC] doit être utilisée pour l’accès à la base de données*

***4.3 Diagramme de classes***

Nom de classe: forme nominale singulier (par contre en base de données : nominale plural)

Nom d’attribut : forme nominale

* Booléen : forme verbale + fait à vérifier

Nom d’objet : forme nominale en relation à la classe auquel il appartient

Nom d’opération : forme verbal (les objets sont le sujet a qui l’opération s’appliq

Nom de paramètre : forme nominale qui désigne un rôle à jouer pour les paramètres

Nom de rôle : forme nominale qui désigne un rôle à jouer pour les objets

Nom d’association : forme verbale (les objets jouent le rôle de sujet de cette forme verbale)

***\*CARDINALITE***

Les cardinalités minimales ou maximas sont composés de constantes entières positives ou \*

**INTERDIT :**  0..n

🡪 Forme correct : 0,4-6,9-\*

*Exemples :*

Nom de classe : *Lampe*

Nom d’attribut : *Alumie*

* Booléen : *estEteinte*

Nom d’objet : *Franck* (classe *Personne*)

Nom d’opération : *appeler*

Nom de paramètre : *communication*

Nom de rôle : *doctor*

Nom d’association : *écoute*

**4.4 Base de données**

Nom de clase : forme nominale au singulier

Nom de relation : forme nominale pluriel définie dans le glossaire, correspond à une ensemble d’identités.

* Si une abréviation est utilisée, il doit être défini dans le glossaire

Nom de colonne : doit correspondre à une forme nominale qui se correspond avec l’attribut ou au concept représenté et qui doit être dans le glossaire. La colonne booléenne peut être appelée pour une forme verbale.

* Si une abréviation est utilisée, il doit être défini dans le glossaire

Nom de clé étrangère : les colonnes correspondant à des clés étrangères doit permet d’identifier l’entité référentiel.

**IMPORTANT : Il faut spécifier**

* Schéma 1 FN : schéma donnée en 1ère forme normale
* Schéma 2 FN : schéma donnée en 2ème forme normale
* Schéma 3 FN : schéma donnée en 3ème forme normale

*Exemples :*

Nom de clase : *Personne*

Nom de relation : *LesPersonnes*

Nom de colonne : *adresse, estArrive*

**4.5 UML Modelio**

*Ecriture :*

Nom de classe : ne peut pas avoir les espaces blancs, il doit commencer pour majuscule et tous les mots individuelles suivant la première devraient commencer pour majuscule aussi.

Nom d’attribut : ne peut pas avoir les espaces blancs, il doit commencer pour minuscule et les mots individuelles suivant la première devraient commencer pour majuscule. Le format pour la définition est le suivant

Visibilité nom [multiplicité] : tipe d’expression = valeur initiale

Nom d’opération : ne peut pas avoir les espaces blancs, il doit commencer pour minuscule et les mots individuelles suivant à la première devraient commencer pour majuscule. Le format pour la définition est le suivant

Visibilité nom (paramètres) : paramètres de sortie

Nom de rôle d’une association : ne peut pas avoir les espaces blancs, il doit commencer pour minuscule et les mots individuelles suivant la première mot doit commencer pour majuscule.

*Normative:*

* Les acteurs et les uses de case peuvent avoir seulement un comportement binaire
* Attributs, associations et opérations ne peuvent pas être abstracts et être une classe au même temps
* Une interface ne peut pas être implémentée deux fois pour la même classe
* Toutes les opérations devraient avoir un nom différent
* Tous les paramètres d’une opération devraient avoir un nom différent
* Une opération privée ne peut pas être redéfinie
* Une opération abstract ne peut pas définir une opération concrète
* Un constructeur ne peut pas avoir paramètres retournées
* Une association doit avoir un unique nom, une autre association ne peut pas avoir le même nom
* Tous les paramètres d’une classe devraient avoir noms différentes. Le nom de tous les attributs devraient être différents aussi
* Un attribut ne peut pas avoir le même nom qu’une association
* Le rôle d’une association doit être associé impérativement avec le rôle de la classe
* Le rôle d’une association ne peut pas être le rôle d’une association pour une autre classe
* Les acteurs peuvent seulement avoir associations avec uses de casse et classes