Introduction

La réussite de tout projet dépend de la qualité de son étude.

De ce fait, l’étape de spécification constitue la case de départ de notre travail. En outre, l’adéquation de tout application à réaliser aux besoins des utilisateurs assurera la réussite de l’application et son utilité future bue les besoin évolutifs des utilisateurs.

Pour atteindre ces objectifs, une vue claire des différents besoins escomptés de notre projet s’aberre utile. Dans ce chapitre, nous commençons par présenter la méthodologie de gestion de projet adaptée, par la suite nous spécifions les besoins fonctionnels et non fonctionnels. L’analyse de ces besoins nous permettra l’identification des différentes fonctionnalités et les acteurs du système.

2.1 choix de la méthodologie de gestion de projet :

Un projet consiste à vouloir réaliser une idée ayant un nouveau caractère. Pour ce faire, l’idée est transformée en objectifs technique (ce qu’il y a à faire), de délai (combien de temps cela prendra pour le faire) et de cout (avec quel budget).

Le choix d’une méthode de développement est une étape cruciale pour la réalisation des logiciels. Elle dépend de la nature du projet et de sa taille. Pour des projets de petite taille et dont le domaine est maitrisé, par exemple, un cycle de vie en cascade s’avère largement suffisant.

Lorsqu’il s’agit d’un projet où les données ne sont pas réunies dès le départ, où les besoins sont incomplets voire floues. Il faut s’orienter vers une méthode itérative ou orientées prototypes. Parmi les méthodes itératives, nous pouvons distinguer les méthodes agile largement utilisées de nos jours à travers le monde.

Une méthode agile est menée dans un esprit collaboratif et s’adapte aux approches incrémentales. Elle assure une meilleure communication avec le client et une meilleure visibilité du produit livrable. Elle permet aussi de gérer la qualité en continu et de détecter des problèmes de plus tôt au fur et à mesure, permettant ainsi d’entreprendre des actions correctrices sans trop de pénalités dans les couts et les délais.

Il y a plusieurs méthodes AGILES et ne s’agit pas de choisir la meilleure méthode parmi celles existantes. Il s’agit plutôt de sélectionner la méthode la plus adaptée à notre projet.

DESCRIPTION DE LA DÉMARCHE DE

DÉVELOPPEMENT

 Un processus simplifié pour les applications Web

 Un processus définit une séquence d’étapes,

partiellement ordonnées, qui concourent à l’obtention

d’un système logiciel

 Processus proposé

 Mi-chemin entre

 UP : cadre général du processus de développement …

 XP : favoriser développement , tests et livraisons

 Principes

 **conduit par les cas d’utilisation**, comme UP

 relativement **léger et restreint**, comme les méthodes

agiles, sans négliger les activités de modélisation en

analyse et conception ;

 fondé sur l’utilisation d’un **sous-ensemble nécessaire**

**et suffisant du langage UML**

La nature de projet en tant que projet évolutif et ayant l'aptitude à s'adapter aux éventuels changements . dont tous les besoins ne sont pas encore totalement identifiés, nous a orientées vers une méthode AGILE et plus particulièrement La démarche « **UP**-**XP » .** que nous détaillerons dans la section suivante.

2.1.1 Présentation de la méthodologie **UP**-**XP :**

L'objectif de la démarche **UP**-**XP** est de pouvoir analyser est concevoir un logiciel orienté objet évolutif et ayant l'aptitude à s'adapter aux éventuels changements que peut connaitre le métier de l'entreprise à l'avenir en plusieurs niveaux : Fiscale, productif, organisationnel, juridique, etc.

Pour bien conduire mon projet et m'assurer du bon déroulement des différentes phases, j'ai opté la démarche UP-XP [2] comme une méthodologie de conception et de développement. Après le choix de la méthodologie, j'ai besoin d'un langage de modélisation UML comme un langage de modélisation.

Mon choix s'est basé sur les points forts de ce langage notamment sa standardisation et les divers diagrammes qu'il propose. Aussi UML présente le meilleur outil pour schématiser des systèmes complexes sous un format graphique et textuel simplifié et normalisé.

En effet UML n'est ni un processus ni une démarche, d'où il fallait choisir une méthodologie de conception et de développement que nous devons l'adopter.

**UP- Unified Process :**

La méthode du Processus Unifié (UP pour Unified Process) est un processus de développement itératif et incrémental, ce qui signifie que le projet est découpé en phases très courtes à l'issue de chacune desquelles une nouvelle version incrémentée est livrée.

Il s'agit d'une démarche s'appuyant sur la modélisation UML pour la description de l'architecture du logiciel (fonctionnelle, logicielle et physique) et la mise au point de cas d'utilisation permettant de décrire les besoins et exigences des utilisateurs.

**XP-eXtreme Programming :**

La méthode XP (pour eXtreme Programming) définit un certain nombre de bonnes pratiques permettant de développer un logiciel dans des conditions optimales en plaçant le client du coeur du processus de développement, en relation étroite avec le client.

L'eXtreme Programming est notamment basé sur les concepts suivants :

? Les équipes de développement travaille directement avec le client sur des cycles très courts d'une à deux semaines maximum.

? L'équipe livraison de versions du logiciel interviennent très tôt et à une fréquence élevée pour maximiser l'impact des retours utilisateurs.

? L'équipe de développement travaille en collaboration totale sur la base de binôme ... ? Le code est tester et nettoyé tout au long du processus de développement.

? Des indicateurs permettent de mesure l'avancement du projet afin de permettre de mettre à jour.

 Description de la démarche

 Etape 1 : Spécification des exigences

 Etape 2 : Spécification détaillée des exigences

 Etape 3 : Réalisation des cas d’utilisation

 Etape 4 : Modélisation de la navigation

 Etape 5 : Conception préliminaire

 Etape 6 : Conception détaillée

**2. Introduction**

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit.

Dans ce chapitre j’ai utilisé le langage UML Unified\_Modeling\_Langage afin de recueillir, d’analysé et d’organisé les fonctionnalités de ma solution.

Au début je commencerai à identifier les différentes actions effectuées par les acteurs du futur système informatisé. Ensuite, j’ai identifié les cas d’utilisation et leur description textuelle.

**2.1 Les Acteurs du système informatisé**

Il s’agit de définir dans ce paragraphe les différents acteurs du futur système et leurs rôles.

Un acteur est une entité qui peut être utilisateur humain, dispositif matériel, ou autres système qui interagissent directement avec le système étudié et qui représentent un rôle bien déterminé.

Pour mon système, j’ai distingué 10 acteurs principaux :

* Agriculteur :
* Grossiste :
* Société d’emballage :
* Usine de stockage :
* Exportateur :
* Importateur :
* Distributeur :
* Détaillant :
* Consommateur :
* Banque
* Transporteur

**2.3 Elaboration du modèle des cas d’utilisation :**

L’étude de cas d’utilisation est très importante en phase d’analyse des besoins, il met en lumière les besoins des différents acteurs et leur associations.

**2.3.1 Diagramme des cas d’utilisation :**

Le diagramme du cas d’utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d’un système logiciel. Un cas d’utilisation est une entité cohérente qui représente une fonctionnalité tout modélisant un service rendu par le système. La figure ci-dessous contient le diagramme de cas d’utilisation qui décrit les fonctionnalités du système.

**2.3.2 Description textuelle des cas d’utilisations :**