1. ANALYSE ET CONCEPTION

I-1 INTRODUCTION

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système. La modélisation est la conception et l’utilisation d’un modèle. Modéliser un système avant sa réalisation, permet de mieux comprendre le fonctionnement du système. C’est également le meilleur moyen de maîtriser sa complexité et d’assurer sa cohérence. Ce chapitre nous aidera à présenter d’abord la méthode d’analyse à utiliser ensuite l’outil de modélisation et enfin finir avec une conception détaillée de notre solution.

I-2 PRESENTATION DE LA METHODE D’ANALYSE

I-2-1 PRESENTATION DES METHODES

Les méthodes d'analyse sont essentielles dans le développement d'applications informatiques, offrant une approche systématique pour comprendre les besoins et contraintes du projet. Elles guident la conception et l'implémentation de solutions logicielles adaptées, en collectant, analysant et spécifiant les exigences du système, ainsi qu'en planifiant et organisant les étapes de développement. Parmi ces méthodes, certaines suivent une approche séquentielle, comme la méthode en cascade, où chaque étape est linéaire et prédéfinie. D'autres adoptent une approche itérative et incrémentale, comme l'Unified Process, permettant des ajustements et des améliorations continus par cycles répétés. En outre, certaines méthodes encouragent la participation des utilisateurs finaux, favorisant la collaboration étroite entre les analystes et les utilisateurs pour comprendre et valider les fonctionnalités du système. De toutes ces méthodes d’analyse, nous pouvons citer :

❖ Méthodes traditionnelles séquentielles : modèle en cascade ou en V;

❖ Méthodes itératives et incrémentales : Unified Process.

I-2-1 METHODES SEQUENTIELLES

Les méthodologies séquentielles ne sont que des approches traditionnelles bien établies dans le domaine du développement d'applications informatiques. Elles suivent une séquence linéaire d'étapes distinctes, où chaque phase est réalisée dans un ordre fixe et avec des résultats définis avant de passer à la suivante. Ces méthodologies accordent une importance particulière à la planification minutieuse, à la documentation approfondie et au suivi précis de l'avancement du projet.

Voici une présentation des points positifs et négatifs de ces méthodologies [7] :

❖ AVANTAGES :

▪ structuration claire : les méthodologies séquentielles offrent une structure bien définie avec des étapes séquentielles, ce qui facilite la planification, la coordination et le suivi du projet ;

▪ documentation approfondie : avec un cahier des charges détaillé et une planification précise dès le départ, Waterfall permet d’avoir une idée précise du budget nécessaire à la réalisation du projet et de la date de livraison.

❖ INCONVENIENTS :

▪ la méthodologie Waterfall ne convient pas aux projets plus larges et/ou complexes ;

▪ flexibilité limitée : les méthodologies séquentielles peuvent être moins flexibles lorsqu'il est nécessaire d'apporter des changements ou des ajustements importants en cours de projet, car revenir en arrière dans les étapes précédentes peut être complexe.

I-2-1-2 METHODES ITERATIVES

Les méthodes itératives sont des approches de développement d'applications informatiques qui se caractérisent par une progression itérative et incrémentale du projet. Elles mettent l'accent sur la flexibilité, l'adaptabilité et la collaboration étroite avec les parties prenantes. L’une de ces méthodologies se trouve être l’Unified Process (UP) ou Processus Unifié en français. Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques [8]. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents niveaux de compétences et à différentes tailles de l'entreprise. Le cycle de développement du processus unifié organise les tâches et les itérations en quatre phases :

❖ l’Inception1 : courte pour estimer, planifier, partager une même vison du problème, et engager les hostilités ;

❖ l’Elaboration : développement de façon incrémentale l’architecture du noyau, les risques et la plupart des besoins identifiés ;

❖ la Construction : élaboration des sous-ensembles exécutables et stables du produit final ; ❖ la Transition : mise à la disposition des utilisateurs de la version bêta. Les méthodes Unified Process offrent plusieurs avantages et inconvénients qu'il convient de prendre en compte :

❖ AVANTAGES :

▪ adaptabilité : l'approche itérative et incrémentale de l'UP permet de s'adapter aux changements et aux évolutions des exigences tout au long du projet.

▪ structuration : l'UP fournit une structure claire et un cadre de travail pour le développement d'applications informatiques ;

▪ collaboration : l'UP encourage la collaboration étroite entre les parties prenantes, favorisant ainsi une meilleure compréhension des besoins et une prise de décision collective ; ▪ qualité du produit final : grâce aux itérations et aux tests réguliers, l'UP permet de développer un produit final de meilleure qualité, répondant aux attentes des utilisateurs [8]. ❖ INCONVENIENTS :

▪ temps et coûts : les itérations supplémentaires dans l'UP peuvent entraîner une augmentation des délais et des coûts du projet, notamment si la gestion des itérations n'est pas bien gérée ;

▪ nécessité de compétences : l'UP nécessite des compétences solides en gestion de projet et en développement logiciel pour être mis en œuvre efficacement.

I-1-2 CHOIX DE LA METHODE

I-1-2-1 LE PROCESSUS 2TUP

Après description et études des différentes méthodes, nous avons retenu le processus 2TUP pour les raisons suivantes :

❖ peu exigeant par rapport à la disponibilité du client ;

❖ la séparation des besoins fonctionnels/Architecturaux ;

❖ permet la modélisation graphique (UML) ;

❖ le besoin de documentation de la conception pour faciliter la maintenance future

2TUP (Two Track Unified Process), est un processus de développement logiciel qui met en œuvre la méthode du processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de trois phases essentielles :

❖ la branche gauche (fonctionnelle) : Capitalise la connaissance du métier de l’entreprise. Elle constitue généralement un investissement pour le moyen et le long terme. Les fonctions du système d’information sont en effet indépendantes des technologies utilisées. Cette branche comporte les étapes suivantes :

✓ la capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs ;

✓ l’analyse.

❖ la branche droite (architecture technique) : Capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l’être en effet indépendamment des fonctions à réaliser. Cette branche comporte les étapes suivantes :

✓ la capture des besoins techniques ;

✓ la conception générique

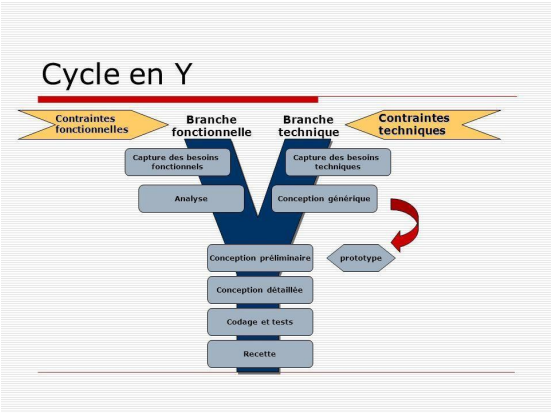
❖ la branche du milieu : À l’issue des évolutions du modèle fonctionnel et de l’architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l’obtention d’un processus en forme de Y. Cette branche comporte les étapes suivantes :

✓ la conception préliminaire ;

✓ la conception détaillée ;

✓ le codage ;

✓ l’intégration.



I-1-2-2 LANGUAGE DE MODELISATION AVEC UML

Pour modéliser notre système, nous avons besoin d’utiliser une méthode. Il a donc fallu associer un processus au langage UML.

LANGAGE + PROCESSUS = METHODE D’ANALYSE.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent. L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientée objet. Fin 1997, UML est devenu une norme OMG.



UML est actuellement à sa version 2.5.1 proposant 14 types de diagrammes séparés en trois catégories à savoir :

❖ les diagrammes de structure ou diagramme statistiques rassemblant :

✓ le diagramme de classe ;

✓ le diagramme d’objet ;

✓ le diagramme de composant ;

✓ le diagramme de déploiement ;

✓ le diagramme de paquet ;

✓ le diagramme de structure composite ;

✓ le diagramme de profils.

❖ les diagrammes de comportement rassemblant :

✓ le diagramme des cas d’utilisation ;

✓ le diagramme états-transitions ;

✓ le diagramme d’activité.

❖ les diagrammes d’interaction ou diagrammes dynamiques :

✓ le diagramme de séquence ;

✓ le diagramme de communication ;

✓ le diagramme global d’interaction ;

✓ le diagramme de temps.

RELATION ENTRE UML et 2TUP

Le processus 2TUP s’appuie sur UML tout au long du cycle de développement, ainsi les différents diagrammes de ce dernier permettent, grâce à leur faciliter et clarté, de bien modéliser le système à chaque étape.

|  |  |
| --- | --- |
| Eléments du processus 2TUP | Diagrammes UML |
| Cas des besoins fonctionnels | * Diagramme de cas d’utilisation * Diagramme de séquence * Diagramme de contexte statique |
| Analyse fonctionnelle | * Diagramme de classe * Diagramme d’état transition |
| Capture de besoins techniques | * Diagramme de cas d’utilisation |
| Conception générique | * Diagramme de déploiement |
| Conception préliminaire | * Diagramme de déploiement * Diagramme de composant |
| Conception détaillées | * Diagramme de classe * Diagramme de séquence * Diagramme de collaboration * Diagramme d’état * Diagramme de composante * Diagramme d’activités |

I-3 PRESENTATION DE L’OUTIL DE MODELISATION

Les logiciels de modélisation de données sont des applications essentielles qui facilitent la création de structures de base de données à l'aide de modèles visuellement attrayants et simples. Ils agissent comme des plans détaillés, guidant la façon dont un projet sera organisé et géré pour atteindre ses objectifs prédéfinis. La modélisation des données est souvent comparée à l'activité d'un architecte créant un plan avant la construction d'une maison. Comme dans le cas d'une construction, un projet informatique nécessite une planification minutieuse qui prend en compte la structure du projet et les besoins définis. Ainsi, plusieurs outils, tels que PowerAMC, StarUML et Diagrams.net, sont disponibles pour améliorer l'efficacité de ce processus. Notre choix s'est porté sur PowerAMC pour ses fonctionnalités et son adaptabilité aux besoins spécifiques de notre projet.



I-4 CONCEPTION DETAILLEE DU PROJET

Cette partie va nous permettre de modéliser les différents diagrammes qui montrent les besoins et les interactions entre les utilisateurs et l’application. Les diagrammes qui vont nous permettre de réaliser notre solution sont les suivants :

❖ Diagramme de cas d’utilisation ;

❖ Diagramme de classes ;

❖ Diagramme d’activité ;

❖ Diagramme de séquence ;

❖ Diagramme de déploiement.

I-4-1 DIAGRAMME DE CAS D’UTILISATION

I-4-1-1 PRESENTATION

Le diagramme de cas d'utilisation est l'un des nombreux diagrammes utilisés dans le domaine de l'ingénierie logicielle pour modéliser les interactions entre les acteurs (utilisateurs) et un système. Il fait partie des premiers diagrammes de notre modélisation car il permet de représenter graphiquement les fonctionnalités principales d'un système. Le diagramme de cas d'utilisation met l'accent sur les interactions entre les acteurs et le système. Il fournit une vision globale des fonctionnalités du système du point de vue des utilisateurs et des différentes actions qu'ils peuvent entreprendre

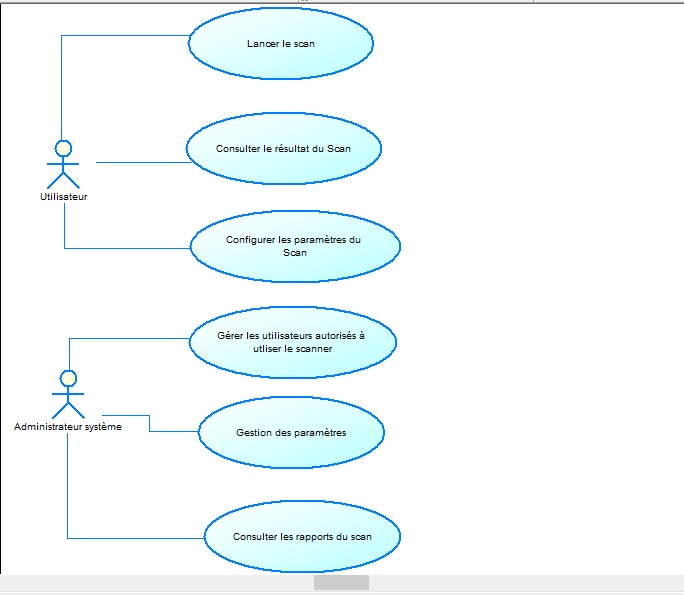
Les acteurs de notre application sont :

❖L’Utilisateur

Administrateur

I-4-1-2 MODELISATION DES DIAGRAMMES

❖ Diagramme des cas d’utilisation « global » Ce diagramme global fournit une vue d'ensemble des interactions entre les acteurs et le système, mettant en évidence les fonctionnalités principales du système d'un point de vue de chaque acteur



I-4-2 DIAGRAMME DE CLASSES

I-4-2-1 PRESENTATION Un diagramme de classes est un outil de modélisation essentiel en UML pour représenter les classes, objets, attributs, méthodes et relations dans un système logiciel. Il permet de visualiser la structure statique d'un système. Selon la complexité, un seul ou plusieurs diagrammes peuvent être utilisés pour modéliser les composants du système. Ces diagrammes sont utiles à différentes étapes de conception. En analyse, ils aident à comprendre les exigences du domaine et identifier les composants. Dans le langage UML, une classe représente un objet ou un ensemble d'objets qui partagent une structure et un comportement commun. Les classes ou instances de classes, sont des éléments de modèle communs dans les diagrammes UML.

I-4-2-2 MODELISATION DU DIAGRAMME

Le diagramme de classes illustré ci-dessous représente toutes entités utilisées dans le système ainsi que les relations entre elles.

