CHAP2 : APPROCHE METHODOLOGIQUE

Dans ce chapitre, nous explorerons le cycle de vie d’un logiciel, englobant l’ensemble des phases du développement d’un site depuis sa conception jusqu’ à sa cessation

2. 1 CYCLE DE VIE D’UN LOGICIEL :

Le cycle de vie d’un logiciel désigne l’ensemble des étapes du développement d’un logiciel, de sa conception a sa disparition. Les étapes typiques comprennent l’analyse des besoins, la planification, la conception, le développement, les tests, le déploiement et la maintenance. Ces étapes peuvent itérative et son conçus pour assurer la qualité la fiabilité et la pertinence du logiciel tout au long de son existence.

2. 2 LES ETAPES DU CYCLE DE VIE D’UN LOGICIEL :

Le cycle de vie d’un logiciel comprend généralement les étapes suivantes :

* L’analyse des besoins : A cette phase, les concepteurs commencent par identifier, rassembler et défini les problèmes, les exigences, les demandes et les attentes des clients actuels lies au produit logiciels. Cela implique la récolte des commentaires des différents parties concernées, y compris les clients, les vendeurs, les professionnels de l’industrie et les programmeurs. Cela donne aux équipes travaillant dessous la prévoyance et le contexte nécessaire pour livrer et produire leur solution logicielle.
* Planification : A cette étapes, l équipes déterminent le budget et les ressources nécessaires pour la réalisation des exigences analyser sans oublier de vérifier les risques encourus.

L équipes fournit aussi les plans nécessaires pour dépasser ces risques. De cette façon la faisabilité du projet est bien étudiée. L équipes se concentre a élaborer la meilleur stratégie pour la mise en œuvre réussit de ce projet.

* Conception : Elle utilise la spécification pour décider des solutions proposées. Elle peut contenu la description des fonctionnalités de l’application. Il s’agit des fonctionnalités préciser lors de la spécification des besoins. La conception peut contenir : la conception des interfaces, la conception des données et la conception de l’architecture matériel
* Le développement :A ce stade du SDLC l’équipe de développement construit les solution logiciel en fonction des décision de conception prise . Des responsables technique et chefs de projets sont responsable du déblocage du processus, la prise de décision et l’assistance technique. Cette étape du travail se termine une fois que les équipes ont packagé et construit leurs codes
* Test : A cette étapes une implémentation logicielle est conditionné et testée pour garantie la qualité. Des tests sont effectués pour garantir que les solutions mises en œuvre répondent au nombre de qualité et de performance. Ce la peut impliqué ou bien des tests unitaire , des tests d’intégration et de bout en bout , une vérification ou une validation ainsi que le rapport ou l’identification des erreurs ou de défauts dans la solution du logicielle .
* Déploiement : Ce la signifie de déployer le logiciel dans l’environnement de production. Les utilisateurs pourront ainsi commencer a l’utiliser
* Maintenance : Cette phase du cycle de vie du logiciel vice a garantir la satisfaction des contenus besoins et le fonctionnement conforme aux spécification initiale. Apres les déploiement et les premiers utilisation par les clients ; la maintenance implique la correction des erreurs découverts suite a des circonstance imprévues et non tester. Elle inclus également la mise a jour des fonctionnalité pour s’adapter au version récentes du logiciel, avec la possibilité d’ajuster de nouvelle option au système .

(Figure : cycle de vie d’un logiciel)

Ce chapitre nous permet de connaitre les diffèrent étapes a suivre lors de la création de notre application. Après avoir défini le cycle de vie de notre application, nous pouvons faire dans le chapitre suivant l’analyse.

Deuxième partie : Analyse et Conception

Chapitre 3 : Analyse

L’analyse web est le processus de suivi et de compréhension de l’utilisation de notre site web par les internautes. Son but ultime est de nous aider à comprendre nos utilisateurs à l’aide de données afin d’optimiser les performances de notre site et d’améliorer l’expérience utilisateur. Dans ce chapitre nous allons effectuer une identification des acteurs, une spécification des taches puis une spécification des scenarios.

* 1. Identification des acteurs :

Un acteur représente un ensemble cohérent de rôles joues par des entités externes (utilisateur humain, matériel ou autre système) qui interagissent directement avec le système étudié et pouvant consulter ou modifier directement l’état du système en émettant et /ou en recevant des messages susceptibles d’être porteur de données.

Les acteurs de nos sites sont les suivants :

-le simple visiteur : Avant de devenir client, l’internaute a la possibilité de consulté le site et la possibilité de s’inscrire pour devenir client. Il s’agit de l’acteur le plus important.

-l’apprenant : il bénéficie des différents services offerts par l’application

-l’enseignant : qui contribue à enrichir le site en apportant des documents, cours

-Client : c ‘est la personne qui cherche les fournisseurs, sélectionne les produits avec les prix .

-Administrateur : personne qui a pour rôle principal de gérer toutes les tables de la base de données

* 1. sSpécification des taches :

Les taches associées à chaque acteur sont :

* 1. Spécification des scenarios :

Les scenarios décrivant chacune des taches définies auparavant sont recapitulés dans le tableau suivant

Chapitre 4 : CONCEPTION

Dans cette phase une nouvelle vue du modèle fait son apparition. Cette vue exprime les modules et les exécutables physique sans aller à la réalisation concrète du système. Elle est bases sur :

* Les diagrammes de séquence : Ils représentent la vue dynamique du système. L’objectif du diagramme se séquence est de représenter les interactions entre les objets en indiquant la chronologie des échanges.
  1. LE MODELE CONCEPTUEL DU SYSTEME :

Pour mettre en place le modèle conceptuel du système d’information, nous allons utiliser la technologie de modélisation UML.

4.1 .1 DEFINITION DU MODELE UML :

Un diagramme UML est une representation graphique,qui s interesse a un aspect precis du modele ; c est une perspective du modele.