

Conception

Introduction

Dans le cycle de vie de notre projet, la conception représente une phase primordiale et déterminante pour produire une application de haute qualité. C'est dans ce stade que nous devons clarifier en premier lieu la vue globale, en décrivant l'architecture générale que nous allons suivre dans la partie réalisation de notre projet. Puis, dans un deuxième lieu nous allons détailler notre choix conceptuel à travers plusieurs types de diagrammes.

I. Conception Générale

Définition cycle de vie :

Le cycle de vie d'une application comprend toutes les étapes depuis sa conception et sa réalisation jusqu'à sa mise en œuvre. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement du logiciel et la vérification de son processus de développement.

L'origine de ce découpage provient du constat que les erreurs ont un coût si élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs le plutôt possible.

Les activités d'un cycle de vie :

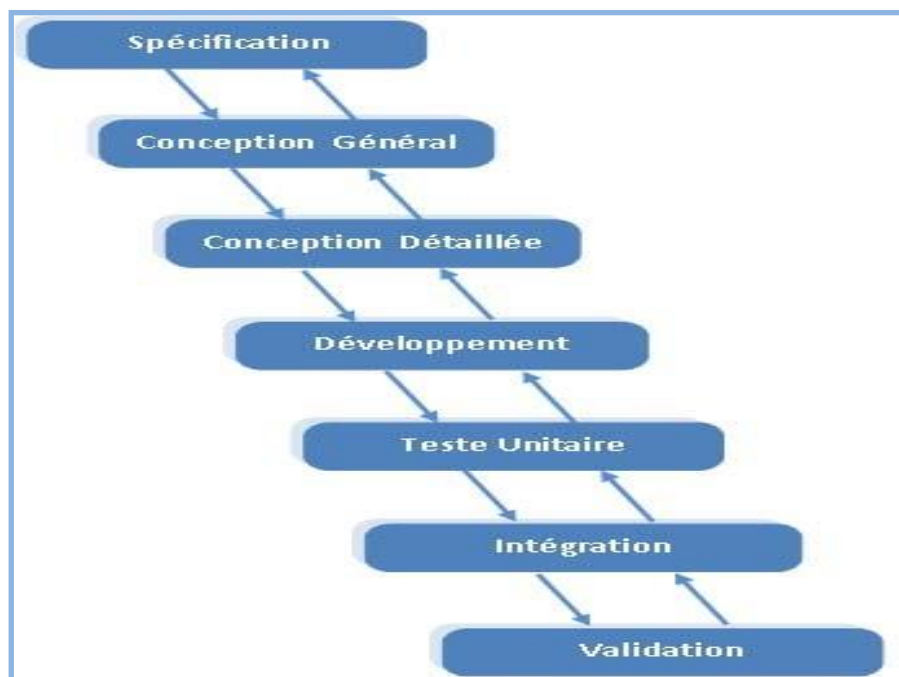
Le cycle de vie suivi pour réaliser un site e-commercial, comprend généralement au minima les activités suivantes :

- **Spécification des besoins:** elle consiste à définir la finalité du projet et son intégration dans une stratégie globale.
- **Conception générale:** dans cette activité, il s'agit de la préparation de l'architecture générale du logiciel.
- **Conception détaillée :** elle consiste à définir précisément chaque sous-ensemble du logiciel.

- **Développement** : (Implémentation ou programmation) il s'agit d'une traduction des fonctionnalités définies dans la phase de conception en langage de programmation.
- **Tests unitaires** : ils permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux normes définies dans la conception.
- **Intégration** : dite aussi tests systèmes, elle consiste à vérifier que le logiciel correspond exactement au cahier des charges du projet en obtenant enfin un manuel d'utilisation bien détaillé aux utilisateurs.
- **Validation** : c'est-à-dire la validation de conformité du site avec les buts spécifiés à la première étape du cycle de vie.

Quelque exemples de modèle de cycles de vie

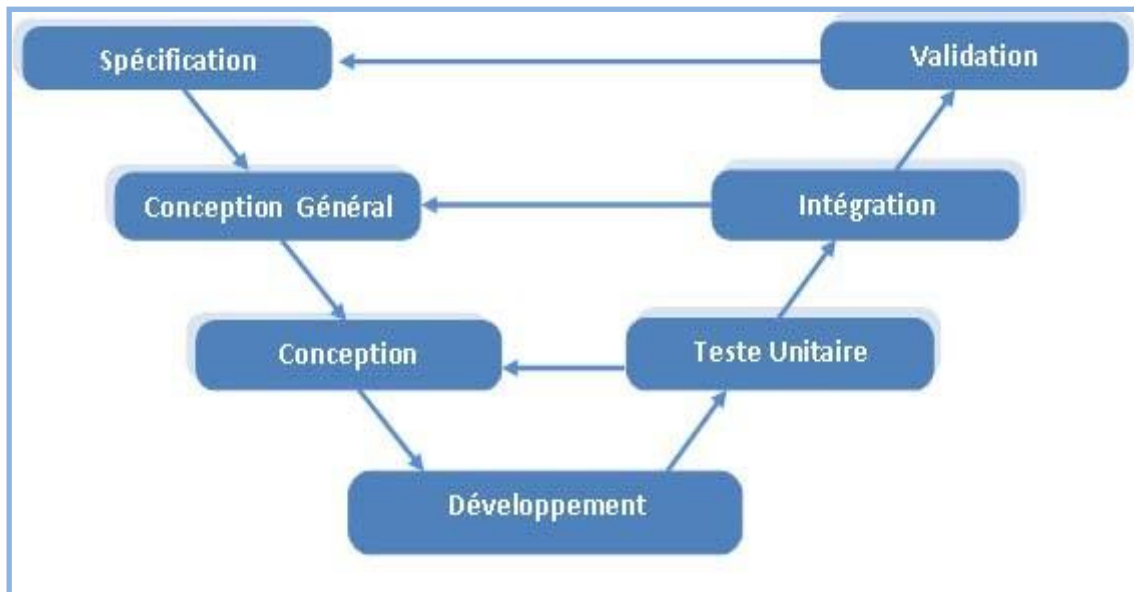
a. Modèle de cycle de vie en cascade



Dans ce modèle le principe est très simple : chaque phase se termine à une date précise en produisant certains documents ou logiciels. Les résultats sont définis à la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés équivalents aux normes. Le modèle original ne comportait pas la possibilité de retour en arrière. Celle-ci a été rajoutée

ultérieurement sur la base qu'une étape ne remet en cause que l'étape précédente, ce qui s'avère insuffisant dans la pratique.

b. Modèle de cycle de vie en V



Le modèle du cycle de vie en V est un modèle conceptuel de gestion de projet, imaginé suite au problème de réactivité du modèle en cascade. Il permet, en cas d'anomalie, d'éliminer le retour aux étapes précédentes tardivement.

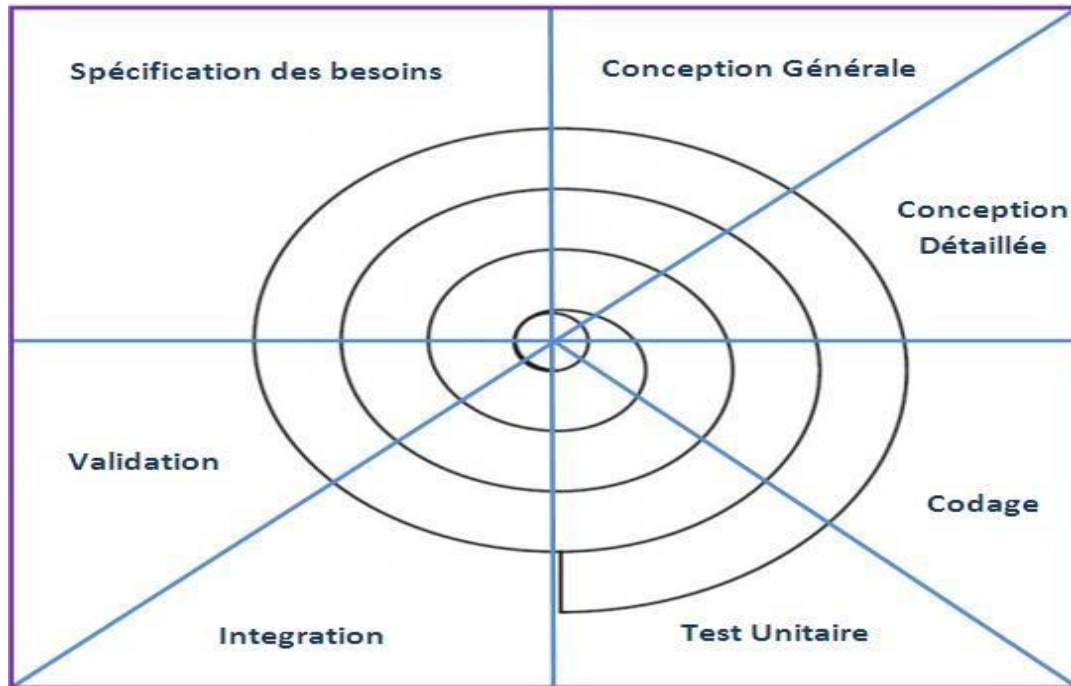
Les avantages du modèle du cycle de vie en V sont les suivants :

- La qualité de la mise en œuvre des tests.
- Modèle éprouvé dans l'industrie.
- Normalisé (ISO-12207, MILSTD-498...)
- Deux types de tâches sont réalisées en parallèle :
Verticalement on prépare l'étape suivante et Horizontalement : on prépare la vérification de la tâche en cours.

Ses inconvénients

- La validation finale par le client très tardive augmente les risques de dépassement de délai et donc l'augmentation du coût.
- Phases séquentielles.
- Rigidité face à une évolution du besoin.

c Modèle de cycle de vie en Spirale



Le modèle en spirale (*spiral modèle*) est un modèle de cycle de vie qui reprend les différentes étapes du cycle en V. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet. Il met cependant plus l'accent sur la gestion des risques que le cycle en V.

I. Conception détaillé

Les diagrammes des cas d'utilisation.

Définition

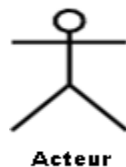
Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système.

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. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Ainsi ces cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

Il ne faut jamais négliger cette première étape pour produire un site web conforme aux attentes des utilisateurs ciblés. Pour élaborer les cas d'utilisation, il faut se fonder sur des entretiens avec les utilisateurs.

Composition du diagramme de cas

Le diagramme de cas se compose de trois éléments principaux :



Un Acteur : c'est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.

Un cas d'utilisation : c'est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie.

Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service. Il représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.

Les relations : Trois types de relations sont pris en charge par la norme UML et sont graphiquement représentées par des types particuliers de ces relations. Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas issu. Une relation simple entre un acteur et une utilisation est un trait simple.

Les acteurs de notre projet

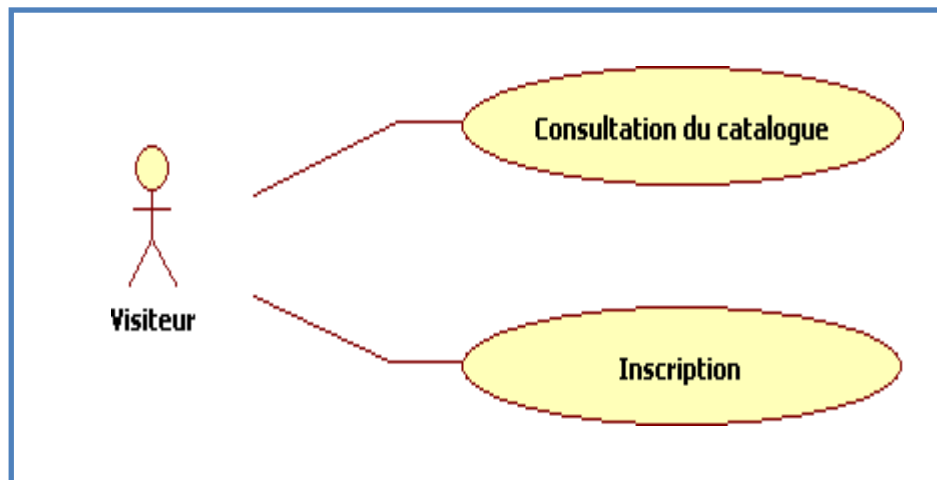
Le visiteur : c'est un individu qui est entrain de fouiller sur le net, cherchant un produit pour l'acheter ou pour avoir une idée sur les modèles et les prix. Jusqu'au ce stade c'est un utilisateur inconnu donc il n'est pas encore un client.

Le Client : cette acteur est un visiteur ayant déjà créer un compte sur notre site, il peut donc suivre le processus d'achat des produits en toute sécurité sachant que notre système doit être l'unique responsable de la confidentialité des données personnelles de ses clients.

L'administrateur : pour les sites web on l'appelle généralement « *le webmaster* ». C'est celui qui assure le dynamisme du site et veille sur les mises à jour des produits, de leurs prix, de leurs disponibilités, de la gestion des paiements et la gestion des livraisons.

Diagrammes de cas d'utilisation de notre site web

a. Diagramme de cas d'un visiteur



Avant de devenir client, un internaute ne possède que la possibilité de consulter le catalogue des produits disponibles dans le stock du fournisseur et la possibilité de s'inscrire pour devenir client sur notre site web.

b. Diagramme de cas d'un client

Après l'inscription, le visiteur devient client. Il est donc apte de continuer toute une procédure d'achat en ligne sur notre site.

c. Diagramme de cas du webmaster du site web

Le terme webmaster de site web désigne communément celui qui est chargé d'un site web. Il gère toute la mise en place technique et parfois la mission éditoriale, il doit gérer au jour le jour la technique et mettre à jour le contenu du site web.

Les activités de notre site web:

La consultation : un catalogue est une foire virtuelle des produits. D'où, il est indispensable de mettre la consultation de ce dernier à la disposition de tous les visiteurs du site sans exception.

L'inscription : après la consultation, et pour passer à la phase d'achat des produits exposés, un visiteur doit devenir client et ce la ne se fait qu'après l'inscription.

L'authentification : c'est une activité principale dans tous les sites de e-commerce. C'est par cette étape que nous allons identifier le client qui est en train de charger son panier et payer sa facture.

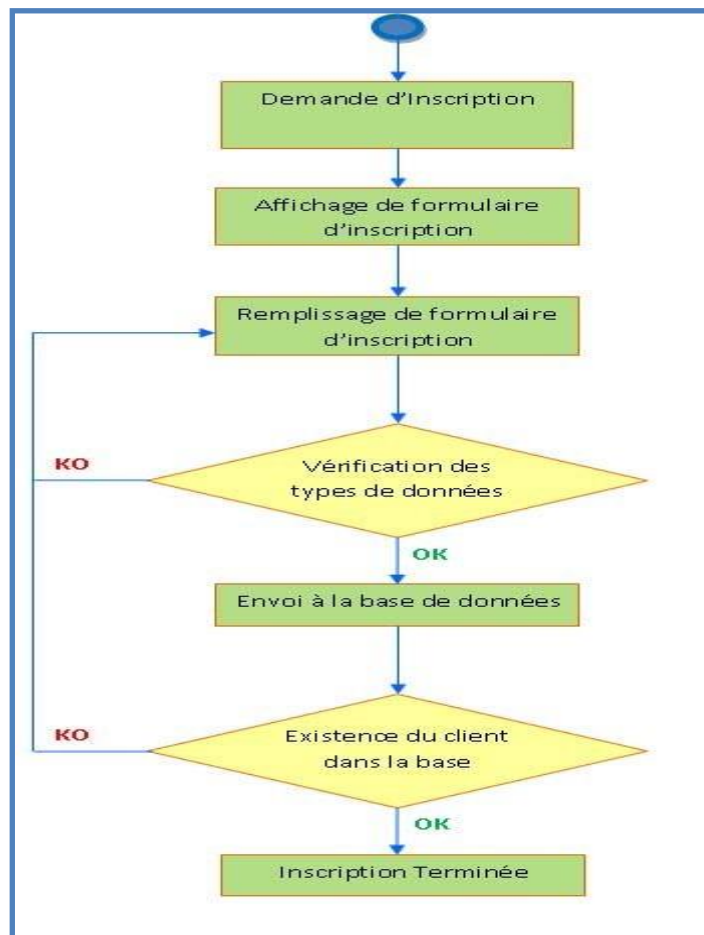
La gestion du panier : suite d'une authentification notre visiteur est maintenant un client qui peut librement ajouter ou supprimer des produits à son panier.

La gestion de stock : cette activité n'est disponible qu'à l'administrateur du site web.

Les diagrammes d'activité de notre site web

a. Diagramme d'inscription

La phase d'inscription est indispensable pour passer d'un simple visiteur du site qui n'a le droit que de consulter les produits et leurs prix à un client qui peut acheter ses articles désirés et payer sa facture en ligne

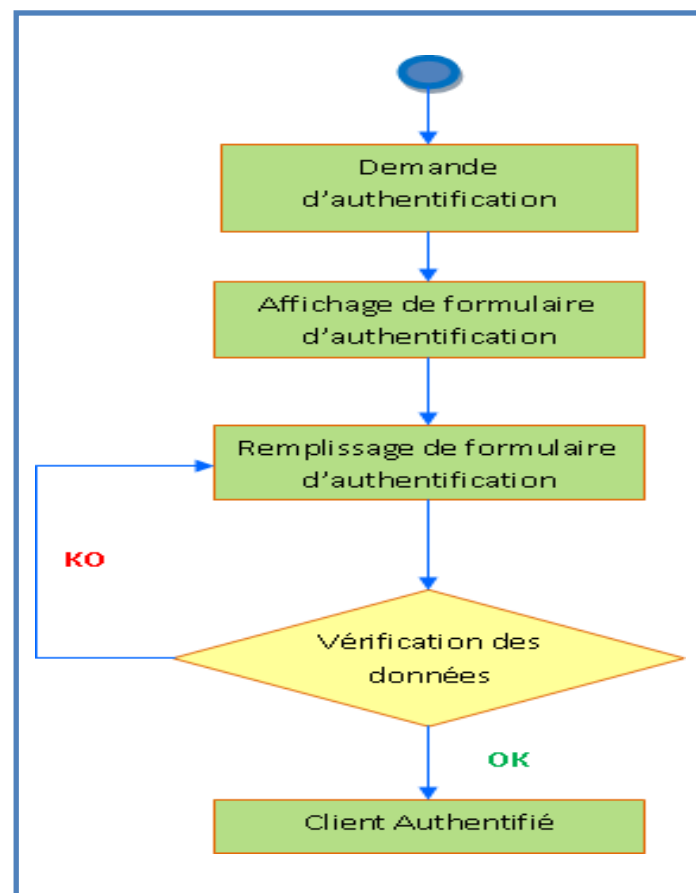


- Le visiteur demande l'inscription.
- Le formulaire d'inscription s'affiche sur l'écran.
- Le visiteur remplit les champs demandés dans le formulaire.
- Le système vérifie les données entrées.
- Si les données sont acceptées, le système les envoie à la base ; si non, il revient à l'étape précédente.
- Le serveur vérifie l'existence du client dans la base.

- Si le client existe déjà, un message d'erreur s'affiche.
- Si le client n'existe pas, l'inscription se termine avec succès.

b. Diagramme d'authentification

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne, ordinateur...), afin d'autoriser son accès aux systèmes, réseaux, applications... Elle permet donc de valider l'authenticité de l'entité en question.



- Le client demande l'authentification en cliquant sur le bouton Login.
- Le formulaire d'authentification s'affiche sur l'écran.
- Le client entre son nom d'utilisateur et son mot de passe.
- Le système vérifie les coordonnées du client sur la base.
- La conformation du succès ou échec est envoyée au client.