Mise en oeuvre : **Création d'un ERP pour une multinationale**

Méthode Waterfall

1 - Spécifications des exigences (Requirement Analysis)

Un document des spécifications des exigences, aussi appelé SRS est un document spécifiant la nature d'un projet / logiciel / application. Il rassemble l'ensemble des exigences, attentes, conceptions et normes pour un projet futur.
Généralement, ce document se compose des éléments suivants :

1.1 - Introduction

1.1.1 - Objectif du produit

L'objectif de ce produit est de créer un système de gestion (ERP) pour un gestionnaire de cliniques vétérinaires. Ce système doit centraliser les différents éléments du logiciel pour fournir une interface utilisateur simple et efficace.

1.1.2 - Public cible

Ce logiciel vise à aider les directeurs de cliniques vétérinaires dans leur gestion et administration.
Mais également les vétérinaires, leur fournissant un planning pour leurs consultations.
Il propose aussi aux clients un système de suivi médical de leurs animaux.

1.1.3 - Portée du projet

Le logiciel est conçu pour maximiser la productivité des vétérinaires et faciliter les processus de gestion des responsables.
Ce système est destiné aux directeurs de cliniques vétérinaires, aux vétérinaires ainsi qu'à leurs clients. Il sera conçu pour de multiples objectifs permettant notamment :

- Aux directeurs de cliniques : de faciliter la gestion de leur clinique. Ils auront la possibilité d'administrer leurs vétérinaires, de s'occuper des commandes auprès des fournisseurs, de leurs stocks, de leurs facturations..
- Aux vétérinaires : Ce système fournira un agenda des consultations permettant aux vétérinaires de s'organiser plus facilement et d'être plus efficaces.
- Aux clients : Le logiciel permettra aux clients de prendre rendez-vous auprès d'un vétérinaire et de suivre le dossier de son / ses animaux.

Ce logiciel vise à être international, une traduction sera donc intégrée.
Le système contient une base de données relationnelle qui stockera des informations sur les cliniques vétérinaires, leurs vétérinaires, les clients, leurs animaux de compagnie, les fournisseurs, leurs produits, les consultations et leurs facturations.

1.1.4 - Convention documentaire

Database	Collections d'informations utilisées par le système
Responsables	Rang d'administrateur, directeur de cliniques vétérinaires
Vétérinaires	Employés des cliniques
Clients	Utilisateur dit 'basique', client des cliniques

1.2 - Description générale

1.2.1 Environnement opérationnel

L'environnement opérationnel du système est :
- Une base de données SQL
- Un système client / serveur
- Un système d'exploitation : Linux
- Plateforme : PHP / JS

1.2.2 Perspective du produit

Un système de base de données stocke les informations suivantes.

Détails du client :
Informations nécessaires sur le client à savoir son nom, prénom, un contact (mail , téléphone), son adresse.

Détails des animaux :
Informations nécessaires concernant les animaux des clients notamment leurs noms, leurs âges, leurs antécédents et précédents consultations, leurs allergies.

Détails des vétérinaires :

Informations sur les vétérinaires (seulement disponible pour les responsables) : nom, prénom, poste, mail.

Détails des consultations :
Informations reprenant le client à l'origine de la prise de consultation, le vétérinaire concerné ou la clinique concernée, ainsi que l'animal qui nécessite la consultation.

Détails des produits:
Informations permettant aux responsables de gérer les différents produits nécessaires. Nom du produit, quantité, prix.

Détails des fournisseurs :
Informations concernant les différents fournisseurs pour les cliniques : adresse catalogue, adresse postale, nom de l'entreprise, contact.

Les différentes caractéristiques du système de base de données peuvent être indiquées dans un modèle entité-relation.

1.2.3 utilisateurs et caractéristiques

Trois rôles ont été définis dans le logiciel :

Responsables :
Utilisateur de type administrateur / directeurs de cliniques. Ils ont accès à la totalité des fonctionnalités et des informations concernant leurs cliniques.

Vétérinaires :
Les vétérinaires peuvent accéder à leurs plannings mais aussi aux dossiers de leurs différents clients. Ils peuvent effectuer des facturations.

Clients :
Ils peuvent réserver une consultation en ligne, consulter le dossier de leurs animaux.

1.3 - Exigences fonctionnelles et systèmes

1.3.1 Cas d'utilisation : Responsables

Le système doit fournir aux responsables les fonctionnalités suivantes :

- Ajouter un nouveau vétérinaire
- Administrer ses vétérinaires : Editer, supprimer
- Consulter les catalogues des fournisseurs
- Effectuer des commandes auprès des fournisseurs
- Accéder à toutes les informations nécessaires (listes) : Vétérinaires, fournisseurs, produits, clients, factures et pouvoir éditer et supprimer ces informations.

1.3.2 Cas d'utilisation : Vétérinaires

Le système doit fournir aux vétérinaires les fonctionnalités suivantes :

- Visualiser et éditer les consultations à venir depuis un planning
- Consulter la liste de ses clients et de leurs animaux
- Accéder aux dossiers des animaux
- Effectuer des factures
- Mettre à jour les stocks de produits et de consultations, facturations

1.3.3 Cas d'utilisation : Clients

Le système doit fournir aux clients les fonctionnalités suivantes :

- Réserver ou annuler une consultation
- Consulter le dossier de son / ses animaux

Il est possible pour les clients n'étant pas encore enregistrés sur le logiciel, de réserver une consultation auprès d'un vétérinaire, qui sera chargé d'enregistrer les clients suite à leurs visites.

1.3.4 Cas d'utilisation détaillés : Clients (enregistré) = Prise de consultation

Description :
L'utilisateur accède au logiciel en ligne, recherche une clinique ou un vétérinaire (qu'il a déjà consulté ou non). Il accède ensuite au planning de ce dernier et réserve une consultation.

Description étape par étape :
Avant l'étape initiale, le client a accédé à la liste des vétérinaires.

1. Le client choisit un vétérinaire parmi la liste.
2. Le système questionne la base de données pour fournir au client le planning du vétérinaire cible.
3. Le client accède au planning et aux créneaux horaires disponibles.
4. Le client sélectionne une date et un créneau horaire pour effectuer sa réservation via un formulaire.

5. Le système enregistre le nom, prénom et contact du client (téléphone et adresse mail) ainsi que l'animal nécessitant la consultation.
6. Le système met à jour le planning du vétérinaire et le notifie par mail qu'une nouvelle demande de consultation lui a été envoyée.
7. Il notifie par mail le client du succès de sa réservation.

1.4 - Exigences non fonctionnelles

1.4.1 Exigences de performances

Les exigences de performances concernent les étapes de mise en œuvre de la base de données mais également les interactions avec cette dernière.

Elle contiennent généralement les éléments suivants :
- Temps de réponse d'une transaction (moyen et maximum)
- Débits pour de données ou de transactions par secondes
- La capacité en nombre d'utilisateurs simultanés ou de transactions que le système peut supporter
- Les dégradations possibles si le système manque de ressources

On peut également retrouver les différents modèles de base de données, nécessaires à la mise en place de cette dernière : Diagramme ER, normalisation

1.4.2 Exigences de sécurité

Les exigences de sécurité sont les éléments assurant la protection des données sensibles sur le logiciel.

Un exemple d'exigence de sécurité : Le système doit être protégé contre tout accès non autorisé.

Contre un dommage grave de la base de données, menant à la perte de données sensibles, il est possible de mettre en place des méthodes de récupération et de restaurations de la base de données à sa version fonctionnelle antérieure par exemple.

1.4.3 Autres exigences non fonctionnelles

Voici une liste des exigences fonctionnelles les plus courantes :

Capacité : les besoins présents et futurs en stockage de votre produit, accompagnés d'un plan qui indique comment votre système s'adapte à un nombre de demandes grandissant.

Compatibilité : les exigences matérielles minimum pour votre logiciel, comme les systèmes d'exploitation et les versions qui peuvent le prendre en charge.

Fiabilité et disponibilité : la fréquence à laquelle vous prévoyez l'utilisation de votre logiciel, ainsi que la durée de panne lors d'une utilisation normale.

Évolutivité : les charges maximales que votre système peut supporter tout en fonctionnant normalement.

Maintenabilité : la façon dont votre application doit utiliser une intégration en continu pour déployer des fonctionnalités rapidement et corriger les bugs.

Utilisabilité : la facilité d'utilisation du produit.

1.4.4 Mesures

Il existe différents types de mesures permettant de répondre plus facilement aux exigences non fonctionnelles.

Mesures de fiabilité :

- Probabilité d'un échec
- Taux d'occurrence de fautes
- Précision des calculs

Mesures de disponibilité :

Proportion de temps durant lesquels le logiciel fonctionne correctement
Mesurée par la durée entre les pannes et par la rapidité de reprise de service :

- Temps moyen entre pannes: MTTF (Mean Time To Failure)
- Temps moyen de réparation: MTTR (Mean Time To Repair)
- Disponibilité = MTTF/(MTTF+MTTR)

Peut mener à des exigences architecturales:

- Composantes redondantes (abaisse MTTF).
- Modificabilité des composantes (abaisse MTTR).

Mesures de sécurités :

Mesure de la capacité du système à résister à des tentatives d'usage non-autorisées et refus de services :

- Taux de succès des authentifications
- Résistance à des types d'attaques connues
- Temps/efforts/ressources nécessaires pour trouver une clé de chiffrage (probabilité de trouver la clé)
- Probabilité/temps/ressources pour détecter une attaque
- Pourcentage de services encore disponibles pendant une attaque
- Pourcentage d'attaques réussies
- Durée de vie d'un mot de passe, d'une session
- Niveau de chiffrement

Mesures de maintenabilité:

Habilité à effectuer des changements rapidement et efficacement :

- Temps moyen pour corriger un bogue, temps moyen pour ajouter une nouvelle fonctionnalité au produit
- Présence, accessibilité, utilitée de la documentation

1.5 - Exigences relatives à l'interface externe

1.5.1 Interface utilisateur

Logiciel front : Logiciel vétérinaire

Logiciel back : SQL

1.5.2 Interfaces matérielles

Navigateur prenant en charge : JS / HTML / PHP

Protocole : HTTP / HTTPS

Appareils : Ordinateurs (responsables, vétérinaires), téléphone (possible pour les clients)

1.5.3 Interfaces logicielles

Logiciel	**Description**
Système opérateur	Linux

Base de données	SQL

1.5.4 Interfaces de communication

Ce projet prend en charge tous les types de navigateurs Web.

2 - Plan de conception du système (Design)

2.1 Architecture logicielle

Lors de la mise en place d'un ERP, il est nécessaire de réfléchir à l'architecture logicielle sur laquelle il reposera.
Une architecture utilisée couramment pour ce type de besoins est l'architecture en modules.
Un ERP est constitué d'un ensemble de modules qui fonctionnent les uns avec les autres.

Ces modules peuvent être séparés par catégories par exemple :

Gestion Achats	Transactions, écritures comptables, gestion des approvisionnements bons de commande et gestion de la production.
Gestion Ventes	Écritures comptables, devis, factures, CRM, e-commerce et e-procurement.
Comptabilité	Comptabilité multinationale, écritures comptables automatisées et gestion multi-devises.
Stockage	Gestion approvisionnement, état des stocks en temps réel, SCM/GCL, mouvements des stocks et entreposage.
Production	Gestion des besoins en fonction des commandes, régularisation des stocks et gestion des plannings de production.
Ressources humaines	Gestion des plannings et gestion de paie.

2.2 Choix techniques

Il est important de définir en amont les différents choix techniques pour cadrer le contexte du projet.

Ce choix concerne les langages de programmation, mais également les solutions de stockage, d'hébergement, d'environnement selon les contraintes et exigences relevées précédemment.

exemple :
Langage de programmation : HTML / CSS / PHP / JS
SGBD : MYSQL

2.3 Modèles de données

Étape importante de la phase de conception du produit, la modélisation permet une visualisation des systèmes de données.

L'équipe de développement va penser et fournir des schémas entités-relations que permettent de mettre en place la base de données.

2.4 Conception détaillée du système (haut et bas niveau)

Au cours de la première, l'équipe élabore le squelette du fonctionnement du logiciel et de l'accès aux informations. Au cours de la seconde, l'équipe développe les parties plus spécifiques du logiciel.
Toutes les fonctions du logiciel sont énumérées et décrites brièvement.

3 - Plan d'implémentation

Un plan d'implémentation est un guide pas-à-pas qui décrit toutes les étapes ou actions à accomplir par votre équipe pour atteindre un objectif.
Le plan d'implémentation se compose généralement de 6 étapes :

Liste d'objectifs pour aider à évaluer les performances et la progression de l'équipe

L'énoncé de portée de projet qui définit les échéances et résultats visés dans les grands axes.

La description des livrables qui sert de guide pour l'allocation des ressources et la délégation des tâches.

L’échéance des tâches qui définit les jalons et planifie l’achèvement du projet. Elle peut-être clarifiée à l’aide d’un diagramme de Gantt notamment.

L’évaluation des risques qui visent à les écarter. Une analyse SWOT peut permettre de mettre en évidence les risques potentiels.

Les rôles et responsabilités des membres de l’équipe. Ils peuvent être mis en place à l’aide d’une matrice RACI.

D’autres éléments peuvent être inclus dans le plan d’implémentation :
- Le responsable de la tâche
- La statut de l’action
- La priorité de l’action
- La progression de l’action

4 - Planning de tests

Le plan de test est un document décrivant l'étendue, l'approche, les ressources et le planning des activités de test prévues.

Il contient :
- Une fiche descriptive qui permet de comprendre pourquoi exécuter des tests et quel fonctionnement va être testé.
- Une description du scope ou out of scope.
- Une analyse des risques, qui permet de couvrir les tests exécutés.
- Les ressources matérielles et temporelles nécessaires.
- Les outils et environnements nécessaires.

Ce plan découle de trois documents :
- La stratégie de test / analyse des risques
- L’analyse fonctionnelle / User stories ou fonctionnalités
- Les documents de critères et exigences (techniques, sécuritaires ..)

Exemple de scénarios de tests :

Type de test	Description	Etape test	Résultat attendu	Statut
Sécurité	Check des règles de mot de passe	Renseigner un mot de passe respectant les	Le mot de passe doit être accepté	Réussi ou échec

		règles		
Usabilité	Vérifier que les liens sont fonctionnels	Cliquer sur les liens de redirection en tant qu'utilisateur	Redirection sur la page cible	Réussi ou échec

Quatres étapes pour mettre en place le plan de tests :

- Mise en place des conditions préliminaires
- Exécution des tests
- Génération des rapports
- Evaluation du risque résiduel

5 - Plan de déploiement & maintenance

5.1 Déploiement

5.1.1 Types de déploiement

Déploiement de base :
Il permet de mettre à jour tous les environnements cibles de manières simultanées sans inclure de processus. Il est dangereux car les applications déployées sont non contrôlées.

Déploiement progressif :
Les applications logiciels vont remplacer progressivement l'ancien logiciel. Il ne conserve pas l'application d'origine.

Déploiement bleu-vert :
Il permet de préserver l'ancien environnement tout en déployant le nouveau en simultanée. Une fois l'application déployée, s'il y a un problème, il est possible de rediriger le trafic vers l'ancien pour qu'il fonctionne de manière optimale

Le déploiement canari :
Il consiste à déployer une application par sous-ensemble. Au début, il est destiné à un petit groupe de personnes, puis il est déployé de manière incrémentielle au travers de versions progressives.

5.1.2 Définition de la méthode

Il faut tenir compte de différents points pour choisir une méthode de déploiement adaptée :

- Quel système, logiciel ou application faut-il déployer ?
- Quels sont les besoins de l'entreprise ?
- Combien d'utilisateurs finaux y aura-t-il ?
- Quels sont les risques du déploiement ?
- Quel outil est utilisé pour suivre le déploiement ?
- Qui fera partie de l'équipe de déploiement ?
- Quand le déploiement sera-t-il effectué ?
- Quand et comment les données existantes du système migreront-elles ?

5.1.3 Planification du déploiement

Cette étape permet de déterminer les fondements du plan de déploiement. Elle inclut les objectifs visés, les indicateurs de performance et les moyens utilisés pour atteindre les objectifs.

5.1.4 Identifier l’équipe de déploiement

Identifier l’équipe nécessaire pour mettre en œuvre la méthode de déploiement ; Faire part de ces ressources à l’équipe de numérisation du CRVS et assurer qu’elle comprend la portée et les objectifs de ce projet.

5.1.5 Calendrier de déploiement

Il est conseillé de diviser le déploiement en tâches, pour une question d’efficacité.
Ces tâches doivent être planifiées dans un calendrier regroupant les échéances de chacune des tâches et leur attribution

5.1.6 Mise à jour et suivi continu

Une fois l'application, le système ou le logiciel déployé, il est recommandé de surveiller le déploiement et de l'optimiser avec les corrections nécessaires en cas d'erreur.

5.2 Maintenance

La maintenance est un entretien régulier d'un site web dans le but de le tenir fonctionnel et de manière optimale.

Il existe 4 types de maintenance :

- **La maintenance corrective** est l'activité consistant à identifier et corriger les défaillances d'un système informatique et à le remettre dans un état opérationnel.
- **La maintenance préventive** est l'ensemble des actions permettant de détecter et de corriger les défaillances potentielles avant qu'elles ne se manifestent.
- **La maintenance curative** vise à remédier aux défaillances en recherchant l'origine du problème afin d'apporter une solution sur le long terme
- **La maintenance évolutive** est un processus continu de suivi et de mise à jour d'une application ou d'un système existant afin de répondre aux besoins changeants des utilisateurs

Principalement, la maintenance se compose des éléments suivants :

- Mise à jour de sécurité
- Mise à jour des thèmes / plug-ins
- Mise à jour du contenu
- Sauvegarde et reprise après sinistre
- Surveillance et optimisation de performances
- Analyse de sécurité
- Gestion des utilisateurs
- Test d'accessibilité
- Surveillance de la disponibilité
- Surveillance du SEO