

Table des matières

I. GENIE LOGICIEL	3
II. MODÈLES DE CYCLE DE VIE.....	5
1. Modèle itératif -Agile.....	5
2. Scrum.....	6
III. CAHIER DE CHARGE.....	7
1. Contexte et définition du problème.....	7
2. Objectif du projet	7
3. Description fonctionnelle des besoins	8
a) Exigences	8
<input type="checkbox"/> Technique de recensement	8
b) Planification des exigences	9
<input type="checkbox"/> Enregistrer les articles	9
<input type="checkbox"/> Enregistrer des documents	10
<input type="checkbox"/> Enregistrer les newsletters.....	10
IV. ARCHITECTURE DU MODELE MCV : CodeIgniter	11

LISTES DES FIGURES

Figure 1: Methode Agile	5
Figure 2: Diagramme de classe	10
Figure 3: Architecture MVC	11

I. GENIE LOGICIEL

Génie logiciel. Le terme **génie logiciel** (en anglais *software engineering*) désigne l'ensemble des méthodes, des techniques et d'outils concourant à la production d'un logiciel, au-delà de la seule activité de programmation.

En d autre terme Nous définissons le **génie du logiciel** comme étant le processus visant la résolution de problèmes posés par un client par le **développement systématique** et **l'évolution de systèmes logiciels** de grande taille et de **haute qualité** en **respectant les contraintes** de coûts, de temps, et autres.

L'objectif du génie logiciel est :

...la résolution de problèmes posés par un client (de manière satisfaisante) ...Il s'agit là du but essentiel du génie logiciel. Rappelons que, dans certains cas, la solution peut être de ne rien développer, si un produit satisfaisant existe déjà (ex : logiciel générique). Il est important de se focaliser dans la résolution du problème du client et non dans l'ajout des options non requises par le client qui ne solutionne en rien le problème.

- *...le développement systématique et l'évolutif...* il s'agit ici de l'application de techniques bien maîtrisées reconnues et standardisées (e.g. IEEE ou ISO) de façon organisée et disciplinée prédisposées à faire *évoluer* un logiciel existant.
- *...systèmes logiciels de grande taille et de haute qualité...* Un logiciel de grande taille est un logiciel qui ne peut être compris par une seule personne et qui exige un travail en équipe et une bonne coordination. Un des défis principaux est d'arriver à subdiviser le travail à accomplir tout en s'assurant que chacune de ces parties fonctionneront harmonieusement ensemble et que le produit final doit répondre aux critères de qualité bien établis.
- *...en respectant les contraintes de coûts, de temps, et autres.* Il est essentiel de souligner ici que les ressources sont généralement limitées, que le bénéfice résultant doit être supérieur aux coûts, que la productivité de l'équipe doit demeurer concurrentielle et qu'une mauvaise estimation des coûts et de la durée du projet peut mener à l'échec du projet.

Activités connexes aux projets de génie logiciel Les projets de génie logiciel font appels à plusieurs activités dont :

- Gestion du projet. Choisir un modèle/processus et des techniques (itératif, incrémental, prototypage) de développement qui permet de diviser le projet en différentes activités, de décider ce qui est inclus dans chaque activité et de décider sur la séquence ou la chronologie d'activités
- . • Définition et spécification des exigences. Définir les exigences fonctionnelles et non fonctionnelles du système à développer qui permette d'établir le cahier de charges.
- Conception (design). Décider comment la technologie disponible sera utilisée pour répondre aux besoins. Ce qui inclut la détermination de ce qui sera réalisé par le logiciel (software) et par le matériel (hardware), la mise au point de l'architecture du système, la définition des sous-systèmes et de leurs interactions, la conception des interfaces des utilisateurs et des bases de données.
- Modélisation (en UML). Créer des représentations du logiciel et de son domaine d'application. Ce qui inclut la modélisation de son utilisation (use case modelling), la modélisation de sa structure (structural modelling) à travers les classes et objets, la modélisation de sa dynamique et de son comportement (dynamic and behavioural modelling) à travers les diagrammes d'activités, d'états-transitions, etc.
- Programmation (en Java). La production du code source de l'application.
- Déploiement (deployment). La distribution et installation du logiciel dans son environnement d'exploitation.
- Assurance de qualité. Tester le logiciel pour évaluer sa la qualité fonctionnelle et structurelle.

II. MODÈLES DE CYCLE DE VIE

Plusieurs modèles de cycle de vie de développement de logiciel sont utilisés pendant le processus de développement logiciel. Chaque modèle de processus suit une série d'étapes uniques en son genre pour assurer le succès du projet. Deux grandes philosophies ou écoles de pensée permettent de les classer :

- La **philosophie classique (Modèles classiques)** adaptée aux gros projets et les projets gouvernementaux qui est stricte et exige des étapes très clairement définies et une Documentation très fournie. Des exemples sont le **Modèle en cascade**, le **modèle en V**, le **prototypage**, etc.
- La **philosophie agile (Modèles agiles)** qui convient aux projets de petite et moyenne taille. Ces modèles sont incrémentaux et itératifs avec des petites et fréquentes livraisons et dont l'accent est mis sur le code et moins sur la documentation. Des exemples sont **Scrum**, **XP** (eXtreme Programming), **UP** (Unified Process), etc.

Pour notre développement nous utiliserons le modèle agile plus précisément la méthode Scrum.

1. Modèle itératif –Agile

Dans le modèle itératif, le processus itératif commence par une implémentation simple d'un petit ensemble d'exigences logicielles et améliore de manière itérative les versions évolutives jusqu'à ce que le système complet soit implémenté et prêt à être déployé. Un modèle de cycle de vie itératif ne cherche pas à faire une spécification complète des exigences. Au lieu de cela, le développement commence par la spécification et la mise en œuvre d'une partie seulement du logiciel, qui est ensuite examinée pour identifier d'autres exigences. **L'idée de base de cette méthode est de développer un système à travers des cycles répétés (itératifs) et en petites portions à la fois (incrémentales).**

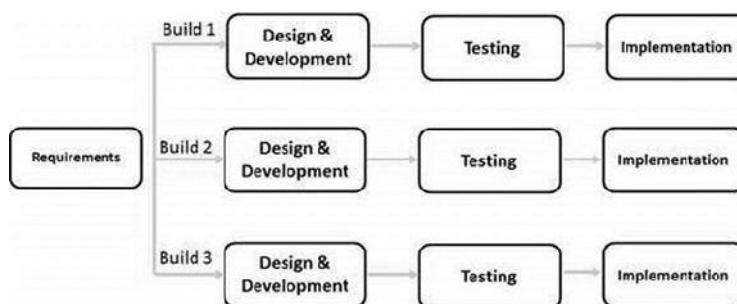


Figure 1: Méthode Agile

Le développement itératif et incrémental est une combinaison de la conception itérative ou de modèle de construction incrémental pour le développement. "Pendant le développement du logiciel, plus d'une itération du cycle de développement logiciel peut être en cours en même temps." Dans ce modèle incrémental, l'ensemble de besoins est divisé en différentes constructions. Au cours de chaque itération, le module de développement passe par les phases d'exigences, de conception, de mise en œuvre et de test. Chaque version suivante du module ajoute une fonction à la version précédente. Le processus se poursuit jusqu'à ce que le système complet soit prêt selon les exigences.

La clé d'une utilisation réussie d'un cycle de vie de développement logiciel itératif est la validation rigoureuse des exigences et la vérification et le test de chaque version du logiciel par rapport à ces exigences dans chaque cycle du modèle. Comme le logiciel évolue au cours des cycles successifs, les tests doivent être répétés et étendus pour vérifier chaque version du logiciel.

2. Scrum

Scrum est un framework qui est utilisé pour implémenter la méthode Agile de développement et de gestion de projet.

Scrum (qui signifie mêlée au rugby) est aujourd'hui la méthode agile la plus populaire. Elle se caractérise par itérations (appelées sprints) assez courts (maximum 1 mois) et un formalisme réduit : rôles (Product Owner, ScrumMaster, équipe), timeboxes (planification de release, planification de sprint, scrum quotidien, revue de sprint, introspection) et artefacts (backlog de produit, plan de sprint, burndown/burnup de release, burndown/burnup de sprint).

On peut comparer la méthode Agile et Scrum à une recette et un régime.

- Avec SCRUM l'estimation de la taille du projet est collective. C'est l'équipe présente qui estime taille et la durée du projet.
- L'estimation se base sur la capacité de l'équipe : la vélocité. La vélocité est généralement déterminée à partir des sprints précédents.
- Une technique pour estimer la durée d'un projet (agile – scrum) est celle des points.

- C'est une technique qui s'inspire du « Count » ou le « Compute » de McConnell. – On sélectionne la story qui semble la plus petite et on lui affecte le poids de 1 – Il s'agit d'estimer les autres stories en comparaison avec la plus petite en y affectant des valeurs de 1 à 10. (La story B vaut 3 fois la story A) – On peut utiliser également la suite de Fibonacci (1, 2, 3, 5, 8, 13, 21...). – Le nombre de sprints est la taille du projet divisé par la vitesse P
- Avec SCRUM l'estimation de la taille du projet est collective. C'est l'équipe présente qui estime la taille et la durée du projet.
- L'estimation se base sur la capacité de l'équipe : la vitesse. La vitesse est généralement déterminée à partir des sprints précédents.
- Une technique pour estimer la durée d'un projet (agile – scrum) est celle des points.
- C'est une technique qui s'inspire du « Count » ou le « Compute » de McConnell. – On sélectionne la story qui semble la plus petite et on lui affecte le poids de 1 – Il s'agit d'estimer les autres stories en comparaison avec la plus petite en y affectant des valeurs de 1 à 10. (la story B vaut 3 fois la story A) – On peut utiliser également la suite de Fibonacci (1, 2, 3, 5, 8, 13, 21...). – Le nombre de sprints est la taille du projet divisé par la vitesse P

III. CAHIER DE CHARGE

1. Contexte et définition du problème

Le laboratoire a été créé et ouvert en septembre 1990. Les premiers matériels ont été mis en place le 17 septembre 1990. Le laboratoire a été immédiatement ouvert à tous les chercheurs-enseignants intéressés. Le laboratoire informatique n'a pas été créé pour procurer aux chercheurs des possibilités de traitement informatisé des textes. Avant le début de l'ère de l'informatique personnelle, les laboratoires informatiques pris en charge des programmes avancés d'études dans le domaine des sciences avec un accès limité à des services spécialisés sur les systèmes mainframe début.

2. Objectif du projet

Le site de laboratoire EEQUAIN est un site qui présente les différents résultats des travaux de recherche à travers de nombreuses publications de documents, des articles, des newsletters. Aujourd'hui, les établissements d'enseignement à tous les niveaux scolaires fournissent des laboratoires informatiques à offrir des services de technologie de l'information à leurs élèves, enseignants et membres du personnel. Bien que l'accent de chaque laboratoire peut varier, les objectifs et les objectifs fondamentaux au sein de laboratoires d'informatique sont les mêmes

quelles que soient les populations qu'ils desservent. Général des services de laboratoire. Un laboratoire d'informatique générale - services rend les ordinateurs de bureau disponibles pour les étudiants et les autres usagers du campus qui ont besoin d'accéder à des applications grand public. Des outils sont fournis pour le traitement de texte, feuilles de calcul et la création de diaporama. Ces laboratoires offrent des postes de travail basés sur du matériel générique et peuvent inclure l'accès aux imprimantes de bureau. L'accès au temps de calcul exige la preuve que l'utilisateur dispose des droits sur le campus. Les véritables objectifs sont l'expérimentation des technologies informatiques de pointe et leur application, au fur et à mesure de leur évolution, à des tâches de recherche, qui doivent se trouver ainsi facilitées, accélérées ou renouvelées.

3. Description fonctionnelle des besoins

a) Exigences

➤ Technique de recensement

Analyse de la concurrence ou Benchmark

Le laboratoire EEQUAIN rencontre des difficultés à toucher un grand public. La présentation des travaux de recherche reste un peu crispée à une audience réduite à savoir ceux qui ont accès directement au laboratoire.

Parmi les principaux laboratoires renommés, on retrouve celui de l'université de Marseille qui est un laboratoire dont les recherches sont ...

L'université de Marseille dispose d'un laboratoire performant qui effectue de nombreuses publications scientifiques par mois. EEQUAIN quant à lui à un rythme de publication plutôt régulier sauf que l'université de Marseille dispose d'un site internet accessible à tous.

Ainsi pour disposer d'une audience plus grande, il faudrait se munir d'un site internet présentant de façon plus détaillée les différentes recherches des travaux effectués.

Pour une meilleure présentation des résultats des travaux de recherches pour attirer un grand public, il est indispensable de disposer d'un site internet. Il permettra d'améliorer le processus de publication des travaux.

b) Planification des exigences

Id	Exigences	Etat	Criticisme	Effort	Stabilité	cible
1	Le système EEQUAIN doit permettre à un membre d'enregistrer les articles	Incorporé	Critique	2j	Stable	0.1
2	Le système EEQUAIN doit permettre à un membre d'enregistrer des documents	Incorporé	Critique	2j	Stable	0.1
3	Le système EEQUAIN doit permettre à un membre d'enregistrer les newsletters	Incorporé	Critique	3j	Stable	0.2
4	Le système EEQUAIN doit permettre à un visiteur de consulter la liste des articles publiées	Incorporé	Critique	2j	Stable	0.3
5	Le système EEQUAIN doit permettre à un visiteur de consulter la liste des newsletters publiées	Incorporé	Critique	2j	Stable	0.2
6	Le système EEQUAIN doit permettre à un visiteur de consulter la liste des publications des documents	Incorporé	Critique	4j	Stable	0.3

- **Enregistrer les articles**

Un article représente en fonction du type des événements organisés par le laboratoire. Enregistrer un article consiste à renseigner le titre de l'article, son type, la date et l'heure, le résumé de l'article. Une fois qu'un membre du système a enregistré un article, il suffit juste qu'il effectue la publication de celle-ci pour la rendre visible par les visiteurs.

▪ Enregistrer des documents

Un document renvoie à un ensemble formé par un support et une information (le contenu). Enregistrer un document revient à renseigner le titre, le ou les auteur(s), le nombre de page, la date de création, sa nature. L'utilisateur du système aura la possibilité d'effectuer une publication de plusieurs documents par catégories.

▪ Enregistrer les newsletters

Une newsletter est considérée ici comme une lettre d'information dont le but est de présenter sous une forme concise et pédagogique la richesse et la diversité des activités du laboratoire à travers des rubriques courtes ciblées sur un chercheur, un projet ou encore une activité de dissémination scientifique, ainsi que de lister les faits marquants ponctuant la vie du laboratoire. Enregistrer celle-ci revient à renseigner le titre, le message, l'image ou les images, la date de création.

Diagramme de classe

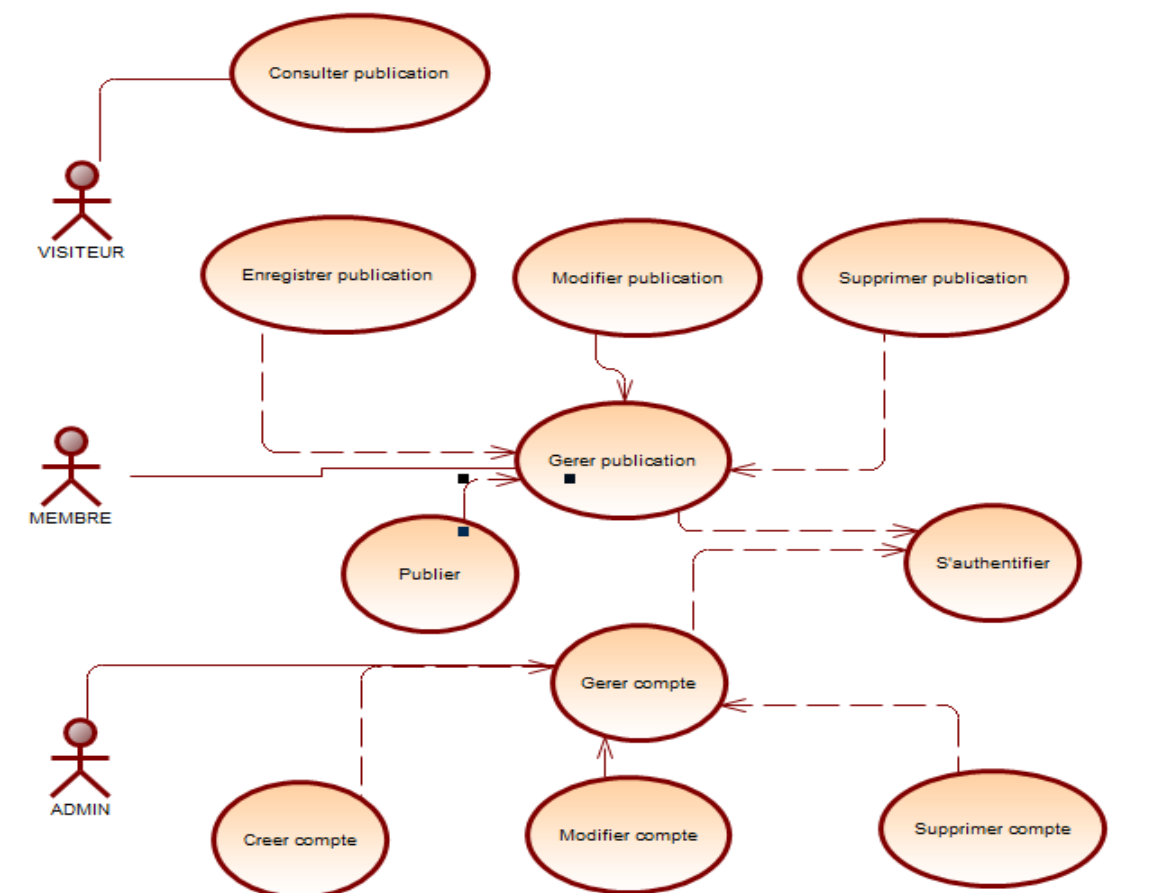


Figure 2: Diagramme de classe

IV. ARCHITECTURE DU MODELE MCV : CodeIgniter

Présentation de Codeigniter

Codeigniter est un Framework php open source basé sur une architecture MVC . Un Framework est un kit qui permet de créer la base de l'application plus rapidement une structure plus solide.

Qu'est-ce que MVC ?

MVC signifie Modèle, Vue et Contrôleur

Architecture MVC

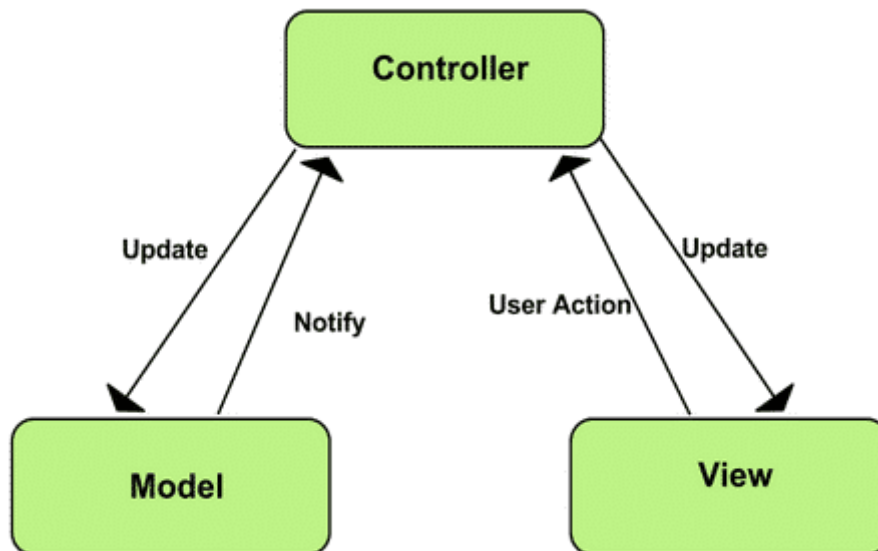


Figure 3:Architecture MVC

Modèle-

- Travail de modèle en tant que back-end.
- Le modèle traite des opérations de back-end.
- Récupérer les données de la base de données et les envoyer au contrôleur

Vues-

- Voir le travail comme front-end.
- Voir les offres avec les opérations frontales.
- Afficher les données et capturer l'action de l'utilisateur
- Envoyer l'action de l'utilisateur au contrôleur

Contrôleurs

Le contrôleur fonctionne comme un intermédiaire entre le modèle et la vue.

Avantage du MVC

- Réduire la flexibilité du code
- Réutilisabilité du code
- Facile à gérer
- Lisibilité du code