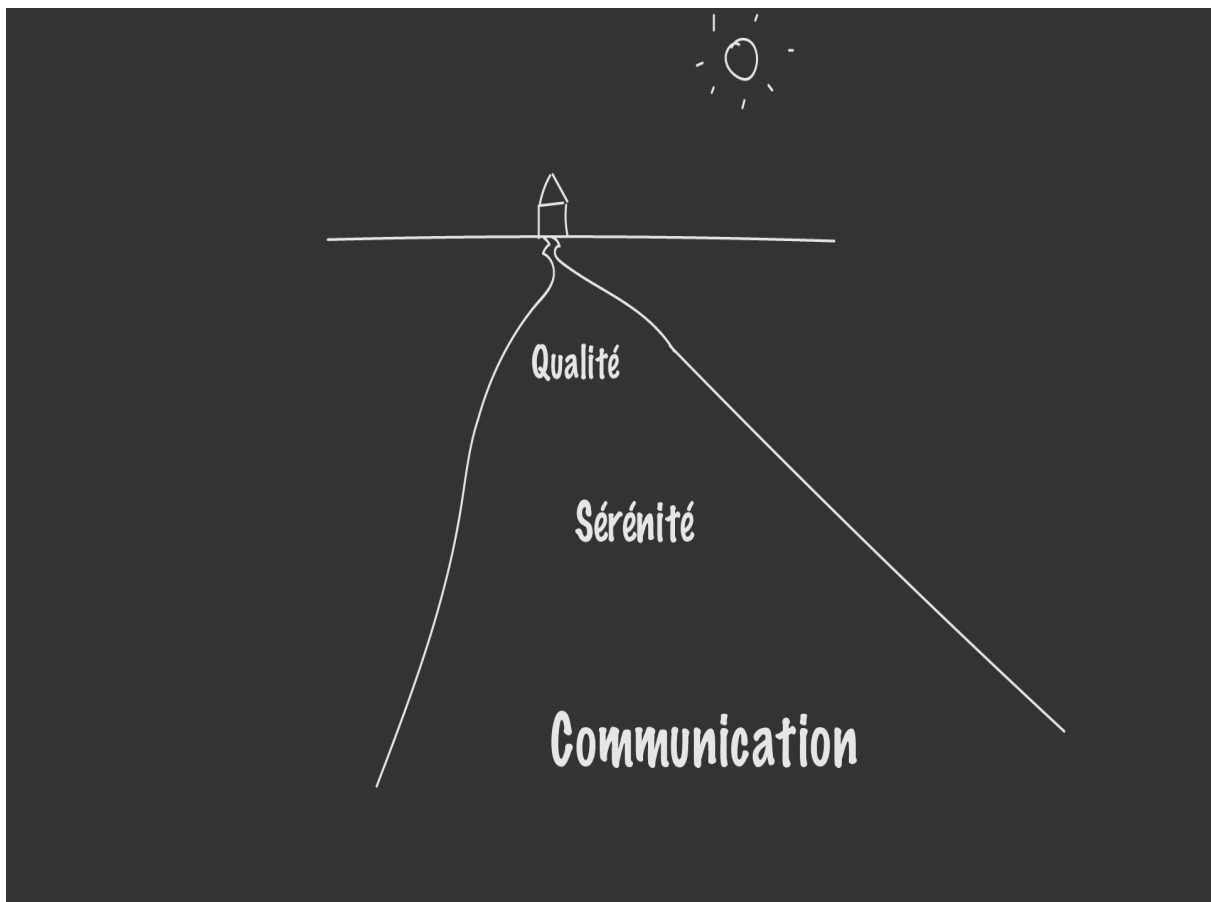


PLANIFICATION DU PROJET

“SharpenSkill”



Planifier un projet a beaucoup d'avantages ! ©OpenClassrooms

Encadré par : **Massinissa HAMIDI**

Élaboré par : **EL Mediouni Sohaib**

Guermane Fadwa

Hasrouf Omar

TABLE DES MATIÈRES

- I. Introduction
- II. Analyse des besoins
 - 1. Définition du périmètre fonctionnel
 - 2. Identification des acteurs
- III. Spécifications fonctionnels générales
- IV. Diagramme de cas d'utilisation global
- V. Adaptation du cycle du développement Scrum au projet
- VI. Spécifications techniques
 - 1. Langage de modélisation utilisé
 - 2. Architecture logicielle
 - 3. Méthodologie agile : méthode Scrum
- VII. Diagramme de classe

I. Introduction

La réalisation de la première phase de la méthodologie Scrum **Sprint 0** qui présente les fonctionnalités de base de l'application, le diagramme des cas d'utilisation globale, et le Backlog qui va nous permettre de planifier les releases (ensemble des sprints).

II. Analyse des besoins

Dans cette partie, nous allons définir d'une manière précise les besoins fonctionnels et non fonctionnels. Tout d'abord, il faut identifier les parties prenantes de l'app.

1. Définition du périmètre fonctionnel :

Évaluer les apprenants sur leurs compétences afin de les orienter vers des parcours de formation personnalisés générés automatiquement.

2. Identification des acteurs :

Visiteur : accéder à la plateforme et découvrir les différentes formations fournies par cette dernière et peut devenir un apprenant

Apprenant : S'authentifier pour accéder à son espace. Disposant le droit de passer des questionnaires afin de déterminer son niveau et accéder à ses formations.

Formateur : Utilisateur disposant du droit d'ajouter une compétence, gérer le questionnaire.

Administrateur : Ayant le droit de mettre à jour un graphe contenant toutes les compétences communes des utilisateurs.

III. Spécifications fonctionnelles générales

Gestion des compétences :

Le formateur ajoute une compétence. Par contre, un administrateur se charge de la mise à jour du graphe global. Ainsi, les formateurs peuvent suggérer des modifications au niveau du graphe des compétences.

Gestion des apprenants :

Un administrateur s'occupe de la gestion des apprenants (ajouter, supprimer). De plus, un formateur peut bannir un apprenant.

Gestion des questionnaires :

Le formateur peut ajouter ou modifier des questions afin de déterminer le niveau des apprenants sur une compétence.

Gestion des formateurs :

Un administrateur peut ajouter ou supprimer un formateur.

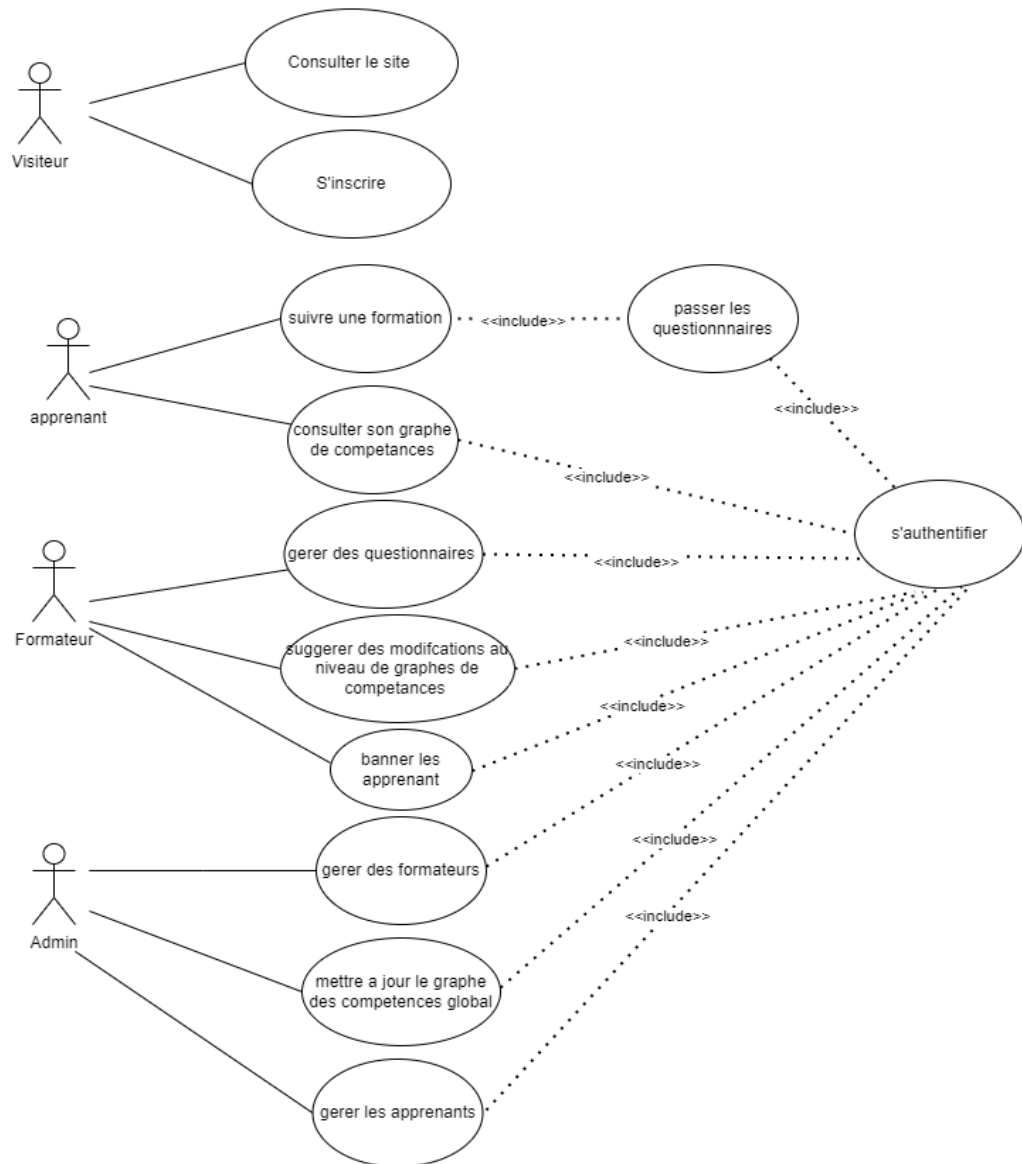
Gestion des formations :

Formateur ajoute des formations au niveau d'une compétence.

IV. Diagramme des cas d'utilisation global

Pour présenter les fonctionnalités de notre app de manière formelle, nous utilisons le diagramme de cas d'utilisation du langage UML.

Ce type de diagramme va nous donner une vision globale du comportement fonctionnel de notre application. Ils sont utiles pour des présentations auprès des parties prenantes du projet.



V. Adaptation du cycle du développement Scrum au projet

1. Répartition des rôles

Product Owner	INSSET
Scrum Master	OMAR HASROUF
L'équipe de développement	SOHAIB EL MADIOUNI FADWA GUERMANE OMAR HASROUF

2. Backlog

La capture d'écran ci-dessous représente le backlog du Sprint 0 :

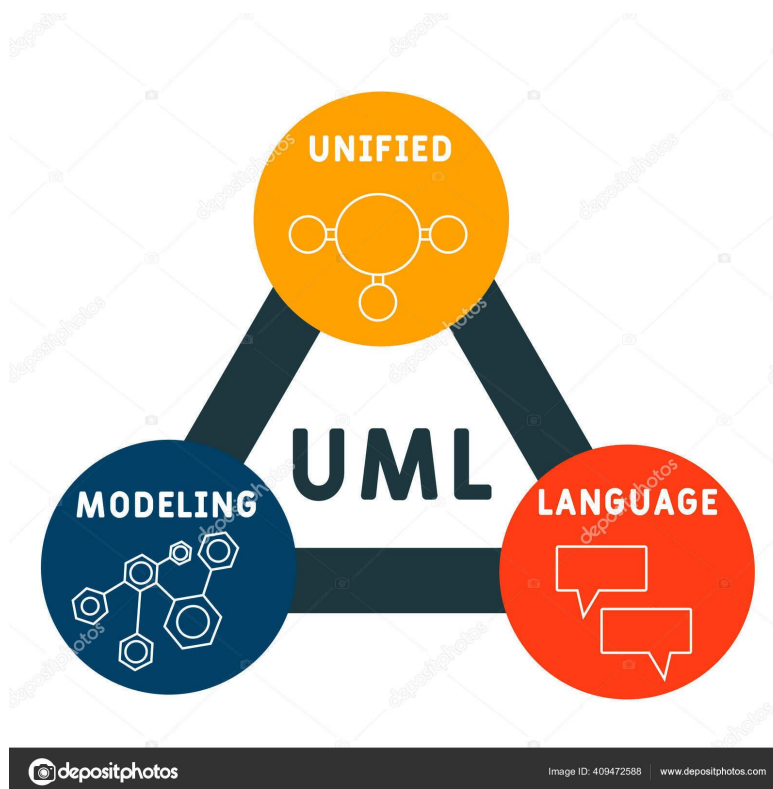
The screenshot displays the 'Agile - Roadmap' interface. At the top, it shows the title 'Agile - Roadmap' with a star icon, a last viewed timestamp 'Vu pour la dernière fois' with a user icon 'FG', and options to 'Inviter / 1' and 'Activité'. A button '+ Ajouter au tableau' is also present. Below the title, a description states: 'Perfect for managing Agile Methodology. The Roadmap hosts upcoming initiatives where you can essentially plan and keep track of all high level milestones.' The interface includes a navigation bar with 'Par défaut', 'Kanban', and 'Current Quarter' tabs, along with an '+ Intégrer' button and an 'Automatiser' button. A search bar 'Recher...' and a filter 'Filtre' are visible. The main section is titled 'Cadrage du projet' and contains a table with the following data:

	Owner	Priority	Domain	Scope defined?	Timeline	Ini
specs fonctionnelles	+2	High	#specs	Yes	févr. 14 - 21	
specs techniques	+2	Medium	#specs	Yes	févr. 14 - 21	
modelisation_V1	FG +2	High	#modeling	Yes	févr. 18 - 21	
+ Ajouter Élément						

VI. Spécifications techniques

1. Langage de modélisation utilisé :

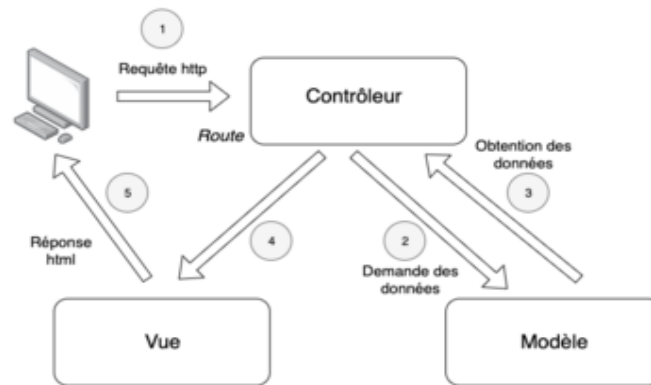
Le langage de modélisation adopté pour notre projet est l'UML. UML, c'est l'acronyme anglais pour "Unified Modeling Language". On le traduit par "Langage de modélisation unifié". La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés diagrammes, qui donnent chacun une vision différente du projet à traiter. UML, nous fournit donc des diagrammes pour représenter notre application à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectués par notre app, etc.



2. Architecture logicielle :

On a choisi une architecture logicielle de notre application qui se base sur le modèle MVC.

En effet, cette approche est l'une des architectures les plus utilisées pour les apps web, elle se compose de 3 module :



Modèle : noyau de l'application qui gère les données, permet de récupérer les informations de la base de données, de les organiser pour qu'elles puissent ensuite être traitées par le contrôleur.

Vue : composant graphique de l'interface qui permet de présenter les données du modèle à l'utilisateur.

Contrôleur : composant responsable des prises de décision, gère la logique métier du code qui prend des décisions, il est l'intermédiaire entre le modèle et la vue.

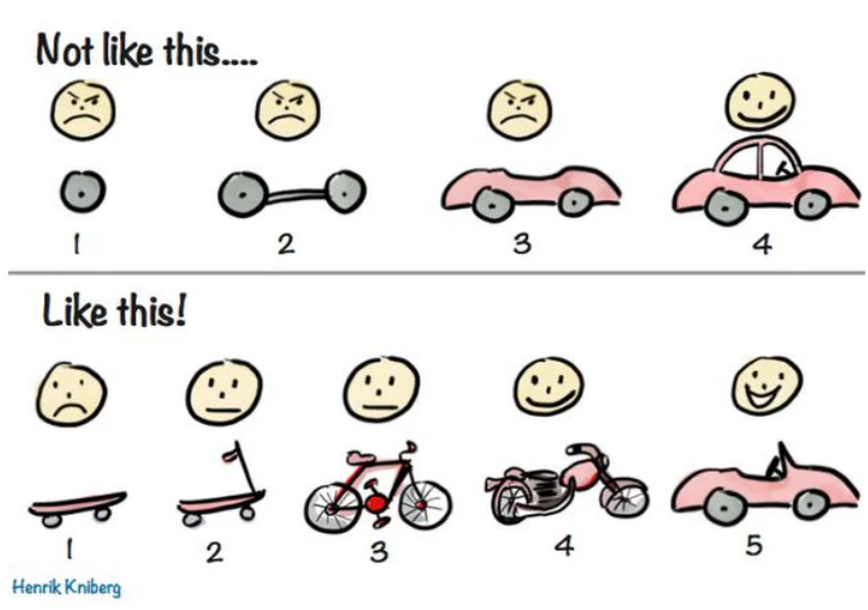
3. Méthodologie agile : Méthode Scrum

C'est le besoin principal de l'application à améliorer au fil des itérations(méthodologie agile) pour arriver à la réalisation finale du projet.

Dans une gestion de projet dite "séquentielle", telle que la méthode en cascade, les différentes phases de développement sont effectuées les unes après les autres dans un ordre précis. Cette méthodologie répond parfaitement aux attentes du client lorsque son besoin est déjà bien défini et délimité.

Dans notre cas, la création d'une application assez innovante qui évolue au fur et à mesure pendant la réalisation nous poussera à choisir une méthodologie agile. Une approche dite "agile", incrémentale (ajouts successifs; mettre en production régulièrement) et itérative (cycles répétitifs; répéter un processus n fois dans le but d'améliorer un résultat), afin d'adapter aux changements imprévisibles. Ainsi, de nous laisser la liberté de faire des changements nous-mêmes lors du projet.

Concrètement, nous allons utiliser la méthodologie la plus connue de projet agile, **Scrum**, pour transformer le cahier des charges en user stories et planifier plusieurs sprints.



Utilisation d'une méthode en cascade VS méthode agile

VII. Diagramme de classe

