

**UNIVERSITE DE YAOUNDE I**

UNIVERSITY OF YAOUNDE I

\*\*\*\*\*

**ECOLE NORMALE SUPERIEURE**

HIGHER TEACHER TRAINING COLLEGE

\*\*\*\*\*



**DEPARTEMENT D'INFORMATIQUE ET DES TECHNOLOGIES**

**EDUCATIVES**

DEPARTMENT OF COMPUTER SCIENCE AND EDUCATIONAL

TECHNOLOGY

\*\*\*\*\*

**ANNEE ACADEMIQUE 2020-2021**

2020-2021 ACADEMIC YEAR

\*\*\*\*\*

**CONCEPTION ET REALISATION D'UNE PLATEFORME NUMERIQUE**

**D'AIDE A L'ENSEIGNEMENT-APPRENTISSAGE POUR**

**L'ENSEIGNEMENT SECONDAIRE AU CAMEROUN**

Mémoire de fin de cycle présenté et soutenu par :

**NJIFEGUEN PEMI Samir Hadi 16Y367**

**Licencié en Informatique**

En vue de l'obtention du

**DIPLÔME DE PROFESSEUR D'ENSEIGNEMENT SECONDAIRE**

**DU SECOND GRADE (DIPES II)**

Filière

**INFORMATIQUE-FONDAMENTALE**

JUILLET 2021

# TABLE DE MATIERES

<b>I Introduction Générale.....</b>	<b>1</b>
1.1. Contexte.....	1
1.2. Problématique.....	2
1.3. Questions de recherche.....	3
1.4. Objectifs de recherche .....	3
1.5. Importance de l'étude .....	4
1.6. Structure du mémoire .....	5
<b>II Revue De La Littérature.....</b>	<b>6</b>
2.1. Etat de l'art.....	6
2.1.1. Typologies de logiciels éducatifs .....	6
2.1.2. Quelques plateformes e-learning existantes.....	7
2.1.2.1. Dans le monde.....	8
2.1.2.2. Au Cameroun .....	9
2.2. Modèles de développement logiciel .....	10
2.2.1. Les modèles classiques ou linéaires .....	11
2.2.1.1. Le modèle en Cascade .....	11
2.2.1.2. Le modèle en V .....	12
2.2.1.3. Le modèle en Y .....	12
2.2.2. Les modèles Itératifs et incrémentaux .....	13
2.2.2.1. Le modèle par Incrément.....	13
2.2.2.2. Le modèle en Spiral.....	14
2.2.3. Les modèles agiles .....	15
2.2.3.1. Le modèle RUP .....	16
2.2.3.2. Le modèle XP.....	17
2.2.3.3. Le modèle Scrum.....	19

2.3. Pratiques Pédagogiques .....	20
2.3.1. Théories de l'apprentissage.....	21
2.3.1.1. Behaviorisme.....	21
2.3.1.2. Cognitivism .....	22
2.3.1.3. Constructivisme.....	22
2.3.1.3. Socioconstructivisme.....	23
2.3.2. Approches pédagogiques .....	24
2.3.2.1. La pédagogie par objectifs .....	24
2.3.2.2. La pédagogie par projet.....	25
2.3.2.3. L'approche par compétences.....	26
2.4. Conception ergonomique des logiciels.....	27
2.4.1. Utilité et utilisabilité des logiciels.....	27
2.4.2. Méthodes d'évaluation ergonomique.....	28
2.4.2.1. Les Méthodes basées sur les utilisateurs .....	28
2.4.2.2. Les Méthodes basées sur l'expertise de l'ergonome .....	29
2.5. Pratiques de recherche .....	29
2.5.1. Population et échantillon d'étude.....	29
2.5.1.1. Population d'étude.....	29
2.5.1.2. Échantillonnage .....	30
2.5.2. Les méthodes de recherche .....	30
2.5.2.1. La méthode quantitative .....	30
2.5.2.2. La méthode qualitative .....	31
2.5.3. Analyse des données .....	31
<b>III Matériel Et Méthodes.....</b>	<b>33</b>
3.1. Choix du cadre conceptuel.....	33
3.1.1. Choix du modèle de développement logiciel .....	33
3.1.2. Choix des pratiques pédagogiques .....	34

3.1.3. Choix de la conception ergonomique .....	34
3.1.4. Choix de la méthode de recherche .....	34
3.2. Méthodologie de travail .....	34
3.2.1. Méthodologie de recherche .....	34
3.2.1.1. Population et échantillon d'étude .....	34
3.2.1.2. Collecte des données .....	36
3.2.1.3. Analyses des données .....	36
3.2.2. Plan de réalisation de Axiom .....	37
3.2.2.1. Exploration .....	37
3.2.2.2. Planification.....	37
3.2.2.3. Construction incrémentale.....	38
3.2.2.4. Mise en production.....	38
3.2.2.5. Maintenance .....	38
3.3. Matériel.....	38
3.3.1. Les logiciels.....	39
3.3.2. Les langages informatiques.....	39
3.3.3. Le matériel.....	39
3.3.4. Les documents.....	40
<b>Bibliographie.....</b>	<b>41</b>

# Chapitre 1

## Introduction Générale

Ce travail s'inscrit dans le cadre de l'intégration des Technologies de l'information et de la communication dans les pratiques pédagogiques de l'enseignement secondaire générale au Cameroun. Dans ce chapitre, nous présenterons le contexte entourant notre étude, le constat fait, les points sur lesquels nous avons orienté notre recherche, les objectifs que nous visons, l'importance de l'étude, le champ dans lequel nous l'avons effectuée et la structure de notre travail.

### 1.1. Contexte

Depuis plusieurs décennies déjà, le monde entier a pris conscience de l'importance de l'intégration des TIC dans l'éducation. En 2007 le Cameroun fait de cela une priorité à travers des initiatives telles que la création du Département d'informatique et de Technologie Educative à l'Ecole Normale Supérieure de Yaoundé I, l'insertion de la discipline informatique dans les programmes de l'enseignement secondaire et la formation des enseignants à l'usage des TIC [Karsenti et al., 2011]. Dès lors, les différentes promotions issues du DITE ont conçu et produit des outils d'aide à l'enseignement-apprentissage à l'instar de DoctorSaam de Magne sur le secourisme et les accidents de l'appareil moteur [MAGNE NKUATE, 2018] et DIPEC de Djoumessi sur la contamination microbienne [DJOUMESSI, 2019]. L'intégration des TIC au Cameroun semble s'effectuer progressivement.

Cependant, en Mars 2020, avec l'arrivée de la pandémie de la Covid 19, le gouvernement camerounais met en place des mesures restrictives incluant la fermeture de tous les établissements scolaires. Cette décision affecte directement la scolarisation de plus de 7,3 millions d'élèves et étudiants [Les Nations Unies au Cameroun, 2020]. L'on est contraint de repenser les stratégies éducatives pour permettre aux élèves de continuer leur apprentissage. Plusieurs méthodes alternatives vont être mises en place comme

l'utilisation du réseau social WhatsApp [Mafouen et al., 2020] et le télé-enseignement. Ces alternatives ont permis dans une certaine mesure de couvrir une partie des programmes officiels dans les classes d'examen. Cependant, pour les classes intermédiaires les enseignements ont totalement été arrêtés. La Covid 19 a mis en mal l'éducation au Cameroun, révélant ainsi le fossé numérique dans le système éducatif [Ntap, 2020].

## 1.2. Problématique

Depuis, les limites de l'enseignement en présentiel se sont fait ressentir :

- Les effectifs pléthoriques dans les salles de classe qui ne favorisent pas le suivi individuel des apprenants.
- Le temps limité alloué aux enseignements qui oblige les enseignants à survoler certaines notions.
- Le manque d'enseignants dans certains établissements qui crée un décalage des enseignements sur l'étendue du territoire, ce qui impact négativement les résultats des élèves.

Les différentes promotions du DITE ont essayé à travers les didacticiels développés, de proposer des palliatifs aux problèmes sus-cités. L'utilisation appropriée de ces outils est à l'issue de la leçon en vue d'une remédiation [Fâtimah, 2020]. Ces outils sont utilisés dans les salles de classe et ne peuvent malheureusement pas constituer un palliatif aux limites de l'enseignement en présentiel.

L'arrivée de la Covid 19 a forcé le système éducatif camerounais à trouver des alternatives à l'enseignement en présentiel avec l'utilisation du réseau social WhatsApp [Mafouen et al., 2020], le télé-enseignement, la mise sur pieds de la plateforme ***distance-learning*** par le MINESEC. Bien que ces alternatives ont permis de combler certaines limites de l'enseignement en présentiel comme les effectifs pléthoriques dans les salles de classe, le manque d'enseignants, la facilité d'accès aux ressources et offrir un enseignement égalitaire à tous les apprenants indifféremment de leurs établissements scolaires, Elles ont cependant posé elles aussi un certain nombre de problème :

- Le suivi des apprenants : elles n'offrent pas la possibilité d'un suivi individuel de la progression des apprenants

- L'évaluation des compétences n'est pas effective : elles ne permettent pas d'évaluer les compétences des apprenants et encore moins de remédier à leurs difficultés.

On se retrouve donc avec des outils dont il est difficile de mesurer l'impact sur le processus d'enseignement apprentissage car ils n'offrent pas de système permettant d'évaluer les résultats de leur usage.

De ce qui précède, nous nous proposons de concevoir une plateforme numérique d'enseignement-apprentissage comblant les limites de l'enseignement en présentiel et résolvant les problèmes posés par les outils d'enseignement à distance.

### 1.3. Questions de recherche

Face au problème soulevé ci-dessus, la question principale que nous nous posons est celle de savoir : **comment concevoir et réaliser une plateforme numérique d'aide à l'enseignement apprentissage adaptée au système éducatif camerounais ?**

Plus spécifiquement, nous demandons :

- Quels sont les aspects à considérer pour développer un tel outil ?
- Comment fournir aux élèves un environnement d'apprentissage interactif facilitant l'acquisition des compétences ?
- Comment fournir aux enseignants un outil de suivi et d'évaluation de leurs enseignements ?
- Comment mesurer l'impact d'un tel outil sur le processus d'enseignement apprentissage ?

### 1.4. Objectifs de recherche

L'objectif général de notre étude est : **concevoir et réaliser une plateforme numérique d'enseignement-apprentissage respectant les normes du système éducatif camerounais et utilisable par les enseignants, les élèves, les parents d'élèves et les politiques.**

Nous avons spécifiquement pour objectif de :

- Concevoir un outil simple qui répond aux critères d'ergonomies logiciel

- Fournir aux élèves un environnement d'apprentissage interactif qui facilite l'acquisition des compétences
- Fournir à l'enseignant un outil de gestion des enseignements, de suivi et d'évaluation des apprenants
- Fournir aux parents d'élèves un outil de suivi des enseignements et des progressions de leurs enfants.
- Fournir au système éducatif camerounais un outil d'uniformisation des enseignements et d'analyse des résultats de la stratégie éducative.

## 1.5. Importance de l'étude

La crise sanitaire due à la Covid 19 qui prévaut actuellement a mis en mal l'éducation dans le monde, en Afrique et au Cameroun en particulier. Les termes utilisés pour nommer cette crise éducative (continuité pédagogique, mesures alternatives, enseignement ou apprentissage à la maison, etc.) témoignent de l'empressement à implanter une approche transposant les habitudes de l'enseignement du présentiel au distanciel. Et bien que plusieurs hésitent à l'admettre, nous entrons de plain-pied dans le domaine de la formation à distance [Karsenti et al., 2020]. Le contexte sanitaire actuel donne déjà à suffisance de l'importance à notre étude. Néanmoins, l'étude que nous allons mener va aboutir à la conception d'une plateforme numérique d'enseignement-apprentissage dont l'impact se fera sur trois (3) plans :

- **Sur le plan académique :** la plateforme à développer sera un outil de gestion du processus d'enseignement apprentissage utilisable par tous les enseignants et tous les élèves, pour toutes les classes et toutes les disciplines, en tout temps et peu importe le lieu. Les apprenants pourront échanger indifféremment de leurs positions géographiques. Elle permettra de palier aux problèmes de l'enseignement en présentiel.
- **Sur le plan socio-politique :** la plateforme permettra d'uniformiser les enseignements sur l'étendue du territoire national, elle sera un outil de riposte face à la pandémie de la Covid 19. Les dirigeants du système éducatif camerounais disposeront d'un outil de mesure et d'analyse en temps réel des résultats de la stratégie éducative ; ils pourront ainsi adapter leur politique éducative à la réalité.



- **Sur le plan scientifique :** notre étude s'inscrit dans le cadre de l'initiation à la recherche des élèves professeurs de l'Ecole Normale Supérieure de Yaoundé I, elle a également pour ambition de développer un outil de collecte de données qui pourra être utilisé comme base de recherche pour la création d'une intelligence artificielle en vue d'améliorer le processus d'enseignement apprentissage.

## 1.6. Structure du mémoire

En plus de ce **premier chapitre** qui tient lieu d'introduction à notre étude, nous organiserons notre travail autour de quatre (04) autres chapitres :

Le **Chapitre 2 : Revue de la littérature** présentera un inventaire des travaux existant en matière de logiciel éducatif

Le **Chapitre 3 : Matériel et Méthode** présentera la méthodologie choisie pour mener à bien notre étude ainsi que le matériel utilisé pour développer notre outil.

Le **Chapitre 4 : Résultats et Discussion** abordera les résultats obtenus après application de la méthodologie et l'appréciation de ces résultats.

Le **Chapitre 5 : Implication sur le système éducatif** présentera les effets de l'utilisation de notre outil sur le système éducatif camerounais.

# Chapitre 2

## Revue De La Littérature

Ce chapitre comprend cinq (5) parties. La première présentera les typologies de logiciels éducatifs et différents travaux qui ont été menés sur la réalisation de plateformes numériques d'enseignement-apprentissage. La deuxième décrira différentes méthodes de développement logiciel. L'outil que nous envisageons réaliser étant un outil pédagogique, la troisième partie abordera les pratiques pédagogiques. La quatrième partie sera consacré aux critères ergonomiques des logiciels. Enfin la cinquième partie décrira les pratiques de recherche utilisées pour un travail scientifique. Toutes ces informations ainsi présentées constitueront les aspects à prendre en compte pour la réalisation d'un outil logiciel pédagogique, utile et facile d'utilisation.

### 2.1. Etat de l'art

#### 2.1.1. Typologies de logiciels éducatifs

Selon l'*Encyclopædia universalis*, un logiciel éducatif est défini comme étant « *un logiciel à but éducatif, conçu pour l'enseignement assisté par ordinateur* ». Il en existe plusieurs types et leur efficacité dépend fortement du choix contextuel et de l'utilisation qu'on en fait [Mukherjee, 2013]. De VRIES propose une typologie de logiciel éducatif fondée sur la fonction pédagogique visée par les enseignants ou les concepteurs et caractérisée sur trois aspects : « *les tâches proposées aux élèves, le point de vue théorique sous-jacent à la conception et la manière dont sont traitées les connaissances* » [De Vries, 2001]. Le tableau suivant en fait une brève présentation.

Type de logiciel	Fonctions Pédagogiques	Tache de l'apprenant	Connaissances
Tutoriel	Présenter de l'information	Lire	Présentation ordonnée
Exercices répétés	Dispenser des exercices	Faire des exercices	Association
Tuteur intelligent	Véritablement Enseigner	Dialoguer	Représentation
Jeu éducatif	Captiver l'attention et la motivation de l'élève	Jouer	
Hypermédia	Fournir un espace d'exploitation	Explorer	Présentation en accès libre
Simulation	Fournir un environnement pour la découverte de lois naturelles	Manipuler, observer	Modélisation
Micro monde	Fournir un environnement pour découverte de domaines abstraits	Construire	Matérialisation
Apprentissage collaboratif	Fournir un espace d'échange entre élèves	Discuter	Construction de l'élève

**Table 2.1** : Classification des outils pédagogiques [De Vries, 2001]

« Un logiciel éducatif peut aussi incorporer plusieurs des fonctions pédagogiques » [De Vries, 2001]. Nous allons recenser les différentes fonctions pédagogiques que notre solution devra intégrer

### 2.1.2. Quelques plateformes e-learning existantes

Les nouvelles technologies de l'information et de la communication ont modifié les apprentissages et leurs méthodes. Le e-learning est apparu ainsi que la formation ouverte et à distance. De nouveaux outils sont nés, et parmi eux, les plateformes pédagogiques. De nombreuses existent, des gratuites, des payantes, des privées, des opensource (libre) [COTINAT, 2017]. La présente étude permet d'analyser quelques plates-formes pédagogiques réaliser dans le monde et au Cameroun.

### 2.1.2.1. Dans le monde

Plus de 250 plateformes de e-learning sont dénombrées à travers le monde à ce jour [FOTCHA, 2008]. Nous présentons ici trois (03) des plus populaires :

- **Moodle :** Moodle (Modular Object-Oriented Dynamic Learning Environment ) est une plateforme d'apprentissage en ligne créé en 2002 par Martin Dougiamas dans le but d'aider les enseignants et formateurs à créer et gérer des cours axés sur l'interaction et la construction collaborative de contenus. Très utilisé par les organismes associatifs et universitaires, elle est développée à partir de principes pédagogiques et permet de créer des communautés d'apprenants autour de contenus et d'activités pédagogiques [Moodle, 2021]. Elle est dotée d'un grand nombre de fonctionnalités : fonctionnalités de communication (chat, messagerie, forum), de planification (agenda), de gestion de contenu (création des leçons, de quizz, atelier de travaux), de travail collaboratif, de suivi des apprenants, etc. Cependant la prise en main de la plateforme peut être complexe du au grand nombre d'outils [DOGBE-SEMANOU et al., 2008].
- **ATutor :** ATutor est un système de gestion de contenu d'apprentissage créé en 2002 pour être accessible et adaptable par les utilisateurs [ATutor, 2020]. Il permet de développer et dispenser des cours en ligne. Il est doté d'un grand nombre de fonctionnalités qui permettent un apprentissage synchrone, un suivi des apprenant, une gestion des contenus, une gestion des évaluations, des échanges entre apprenants, etc. Cependant le fait que la plateforme ait autant de fonctionnalités rend difficile sa prise en main [COTINAT, 2017].
- **Google Classroom :** Google Classroom est une plateforme d'apprentissage gratuite créé en 2014 par l'entreprise américaine Google. Son but est de simplifier la création et la diffusion de cours et d'exercices [Google Classroom, 2020]. Elle propose aux enseignants et aux élèves une base d'échange ayant pour objectif de permettre de coordonner les devoirs, d'augmenter la collaboration et d'optimiser la communication. L'enseignant peut ainsi manager ses cours, créer du contenu, donner des devoirs, noter les travaux et transmettre des feedbacks aux étudiants. Cependant la plate-forme peut être difficile à assimiler si on ne connaît pas déjà les produits Google, on note également l'absence de tests automatiques ou de quizz [Szilas & Venni, 2020].

### 2.1.2.2. Au Cameroun

L'espace numérique camerounais regorge d'un certain nombre de plateformes e-learning qui sont pour la plupart utilisées par des institutions universitaires. Il s'agit notamment de :

- **E-learning FSEG UnivMaroua** : « <https://univ-maroua.com/fseg/edu/> » qui est la plateforme e-learning de La Faculté des sciences économiques et de gestion de l'Université de Maroua. Elle est reconnue comme l'un des plus importants sites académiques du Cameroun dans le domaine économie-gestion en matière d'enseignement et de recherche. La plateforme E-learning de la FSEG de l'Université de Maroua offre à tout son staff et ses étudiants la possibilité de continuer leur formation à distance.
- **elearning.esstic** : « <https://elearning.esstic.cm/> » qui est la plateforme e-learning de l'Ecole Supérieure des Sciences Techniques de l'Information et de la Communication (ESSTIC). Elle permet aux enseignants de déposer les supports de cours et aux apprenants de les télécharger.
- **MineseC Cmr Distance Learning** : « <https://mineseC-distancelearning.cm/> » qui est la plateforme e-learning développée par le Ministère des Enseignements Secondaires du Cameroun pour permettre aux élèves de suivre les cours en temps de Covid 19. Les cours sont déposés sur la plateforme sous forme de média vidéo.

Dans ces différentes plateformes, on note un manque d'interaction. Elles ne sont pas dotées de fonctionnalités de suivi des apprenants, d'évaluation et de communication. Elles permettent de déposer et de consulter des cours

Au Département d'Informatique et des Technologies Educatives (DITE) de l'Ecole Normale Supérieure de Yaoundé, des travaux menés par Fatimah ABOUBAKAR ont conduit à la réalisation de **EasyLearn** qui est une plateforme facilitant l'enseignement apprentissage [Fâtimah, 2020]. Dans ses travaux, elle part du problème qu'ont les enseignants à suivre et à évaluer les compétences de leurs apprenants et propose donc un outil qui est pour l'enseignant un outil pédagogique et pour l'élève un outil lui permettant d'acquérir et d'évaluer ses compétences. EasyLearn met l'accent sur la conception de contenus pédagogiques interactifs.

## 2.2. Modèles de développement logiciel

Le développement logiciel est un Ensemble de processus partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant [Gérard, 2008]. Le développement logiciel ne se résume donc pas à la seule phase de codage mais impose d'effectuer un certain nombre d'étapes appelées **cycle de vie de développement du logiciel** [Oriat, 2007]. Ces étapes sont :

- **L'expression des besoins** : C'est la description informelle des besoins exprimés par l'utilisateur. Cela permet de définir le cahier des charges [Yannick, 2007].
- **La spécification des besoins** : C'est la description formelle du problème par l'équipe de développement. Cela permet de définir le « **quoi** » en fonction du cahier des charges [Yannick, 2007].
- **L'analyse et la conception** : cette phase permet de définir « **comment** » résoudre le problème. La conception est souvent découpée en deux phases : la conception générale et la conception détaillée [Yannick, 2007].
  - La conception générale consiste à d'écrire la structure générale du système [Oriat, 2007].
  - La conception détaillée du système consiste en une description des composants, des algorithmes et des structures de données [Oriat, 2007].
- **Le développement** : cette phase est consacrée à la Production du code en se référant à l'analyse et la conception mais sans avoir besoin de remettre ceux-ci en question. Elle permet aussi la mise en place des tests unitaires [Yannick, 2007].
- **Le test / recette** : phase de validation par le client des fonctionnalités du système [Yannick, 2007].
- **Le déploiement et la maintenance** : phase au cours de laquelle on procède à l'installation (manuelle ou automatisée), la configuration, la mise en production du produit, et la formation des utilisateurs [Yannick, 2007]. Une fois le logiciel déployé il faut le maintenir afin de corriger les défauts (Maintenance corrective), d'ajouter d'éventuelles fonctionnalités (Maintenance évolutive) ou de l'adapter à une nouvelle plateforme (Maintenance adaptative) [Oriat, 2007].

Dans certains cas, ces activités sont regroupées, mais elles sont toujours présentées et se déroulent presque toujours dans l'ordre précité [Yannick, 2007]. Toutefois, il est possible d'organiser le déroulement de ces phases de différentes manières

: en **modèles de développement**. Ainsi, Les modèles de développement ou modèles de cycle de vie décrivent à un niveau très abstrait et idéalisé les différentes manières d'organiser la production du logiciel [Lonchamp, 2015]. Les étapes, leur ordonnancement, et parfois les critères pour passer d'une étape à une autre sont explicités [Lonchamp, 2015]. En fonction des principes sur lesquels ils reposent, On classe généralement les modèles de développement logiciel en trois (03) catégories :

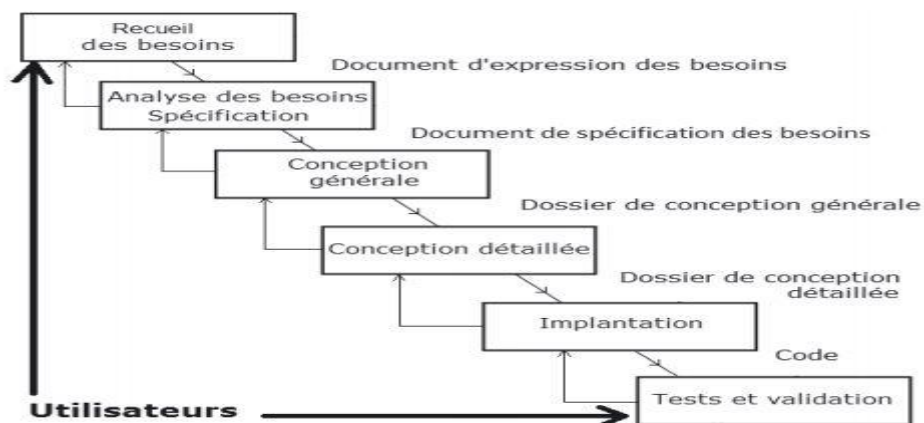
- **Les modèles classiques ou linéaires**
- **Les modèles itératifs et incrémentaux**
- **Les modèles agiles**

### **2.2.1. Les modèles classiques ou linéaires**

Dans ces modèles, un accent particulier est mis sur le séquençement des étapes pour au final produire le logiciel attendu. On distingue plusieurs modèles basés sur l'approche linéaire. Nous présentons brièvement les modèle en cascade, en V et en Y.

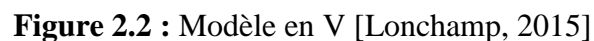
#### **2.2.1.1. Le modèle en Cascade**

Ce modèle est proposé en 1970 par Winston Royce et est hérité des méthodes classiques d'ingénierie [Lonchamp, 2015]. Dans le modèle en cascade on effectue les différentes étapes du logiciel de façon séquentielle. Les interactions ont lieu uniquement entre étapes successives [Oria, 2007]. À chaque étape, il y a production d'un « livrable ». Le livrable est vérifié, s'il est approuver le passage à la phase suivante est effectué, sinon il y a retour pour la correction de l'étape précédent [Yannick, 2007]. Les utilisateurs interviennent au début du processus pour définir les besoins, et à la fin du processus pour valider le système suivant les besoins exprimés au début.



Ce modèle a l'avantage d'être simple à comprendre et à utiliser [Lonchamp, 2015]. Il impose des tests entre chaque phase, ainsi que la définition des tâches et des livrables. Par contre il ne s'adapte pas au changement des besoins du client.

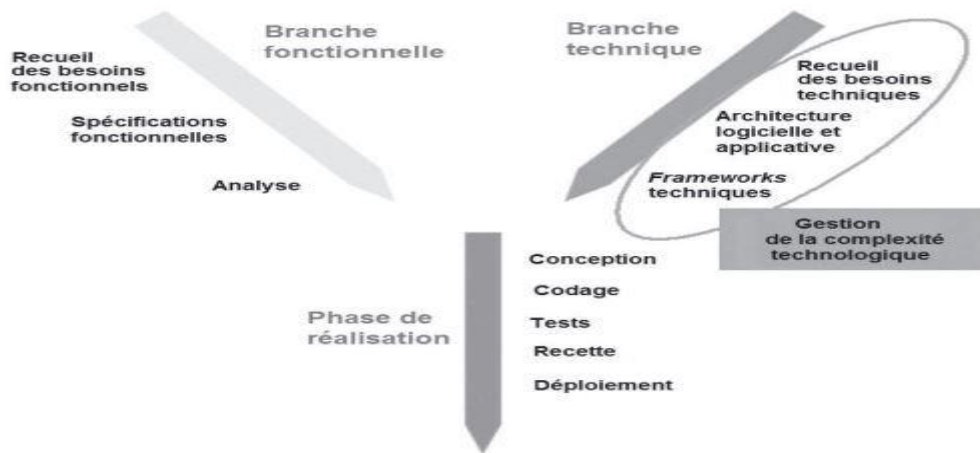
Ce modèle est proposé en 1984 par Mc Dermid et Ripkin en amélioration du modèle en cascade qui présente un manque de réactivité (c'est uniquement lors du test qu'on s'aperçoit qu'il y a un problème) [Yannick, 2007]. Le modèle en V met en évidence la complémentarité des phases menant à la réalisation et des phases de test permettant de la valider. Les tests sont préparés tout au long des phases menant à la réalisation et exécutés en fin de processus [Lonchamp, 2015].



### 2.2.1.3. Le modèle en Y

Ce modèle une autre variante du modèle en cascade qui distingue deux branches : l'une pour les besoins fonctionnels et l'autre pour les besoins techniques [Lonchamp, 2015]. Les deux branches fusionnent ensuite pour la conception du système. Ce est adapté aux projets technologiquement innovants car il permet de lever au plus tôt les incertitudes liées aux technologies à mettre en œuvre [Lonchamp, 2015].





**Figure 2.3 : Modèle en Y [Lonchamp, 2015]**

Toutes les approches linéaires supposent qu'il est possible de spécifier correctement et exhaustivement les besoins en début de processus et que ces besoins restent stables tout au long du processus de développement [Lonchamp, 2015]. Or il advient très souvent que les besoins du client évoluent, comme l'a soulevé [Rota, 2009] « *les plans initiaux sont finalement toujours modifiés et les besoins évoluent en permanence pour répondre aux changements du marché* ». Malheureusement les modèles linéaires ne prennent pas en compte les changements qui interviennent au cours du processus de développement. Ils ne sont pas adaptés pour des projets évolutifs.

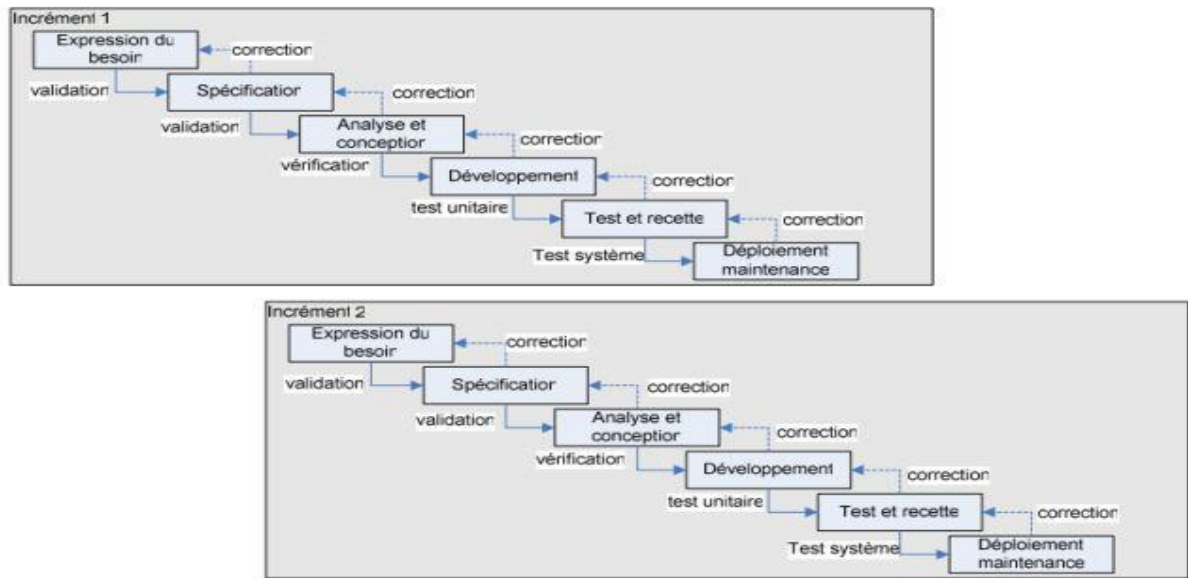
### 2.2.2. Les modèles Itératifs et incrémentaux

Dans ces approches, le logiciel est développé en plusieurs itérations. D'après [Lonchamp, 2015], cela revient à répéter des mini-processus de développement, plus ou moins complets. Chaque itération correspondant au **raffinement** d'un développement précédent ou à **l'ajout** d'un incrément supplémentaire (d'où l'expression « itératif et incrémental ») [Lonchamp, 2015]. Nous présenterons ici le modèle par incrément et le modèle en spirale.

#### 2.2.2.1. Le modèle par Incrément

Le modèle par incrément consiste à découper le logiciel en incrément et à sélectionner successivement plusieurs **incréments**. Un incrément du logiciel est un sous-ensemble du logiciel complet, qui correspond à un petit nombre de fonctionnalités [Oriat, 2007]. Pour chaque incrément, on réalise l'analyse, la conception, l'implémentation et la

validation, puis on livre la version du logiciel. On recommence ainsi jusqu'à obtenir un logiciel complet. On procède donc par versions successives, chaque version donnant lieu à une livraison.

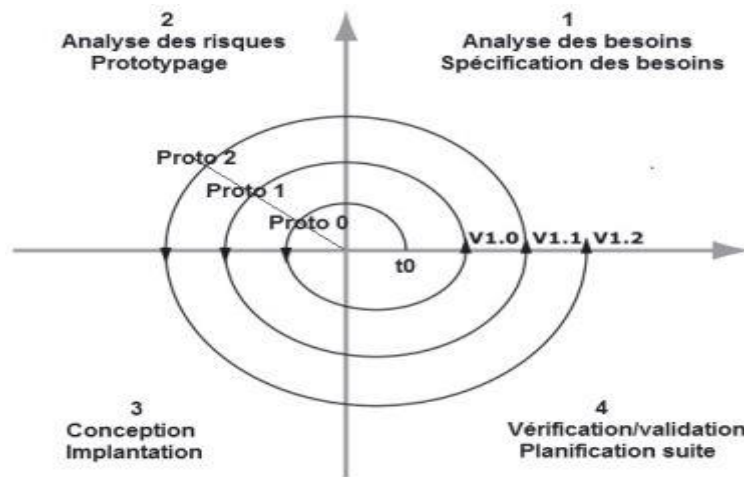


**Figure 2.4 :** Modèle par incrément [Yannick, 2007]

Les avantages de ce modèle sont qu'il permet de révéler certains problèmes de façon précoce, on a rapidement un produit que l'on peut montrer au client. Cependant, après le développement des différents incréments on remarque une difficulté d'intégration de ceux-ci [Yannick, 2007].

#### 2.2.2.2. Le modèle en Spiral

Le modèle en spirale est un modèle itératif, où la planification de la version se fait selon une analyse de risques. L'idée est de s'attaquer aux risques les plus importants assez tôt, afin que ceux-ci diminuent rapidement [Oria, 2007]. D'après [Lonchamp, 2015] chaque itération (ou cycle de la spirale) correspond à une séquence de quatre phases : l'analyse et la spécification des besoins, l'analyse des risques, la conception et le développement, la vérification/validation et la planification du cycle suivant.



**Figure 2.5 : Modèle en Spirale [Lonchamp, 2015]**

L'avantage du modèle en spirale est qu'il permet d'accélérer la réduction du risque au cours du développement du logiciel : les risques sont importants lors des premières itérations, et diminuent lors des dernières itérations. Bien que ce modèle soit très avantageux et s'adapte aux changements, il s'avère difficile à mettre en œuvre dans les projets de grande envergure [Yannick, 2007].

Les modèles Itératifs et Incrémentaux permettent de développer en premier les fonctions primordiales ou les plus risquées. Les clients peuvent réagir après chaque itération. Cependant on note des difficultés de gérer les besoins (« explosion des besoins ») et des difficultés à définir les incréments [Lonchamp, 2015].

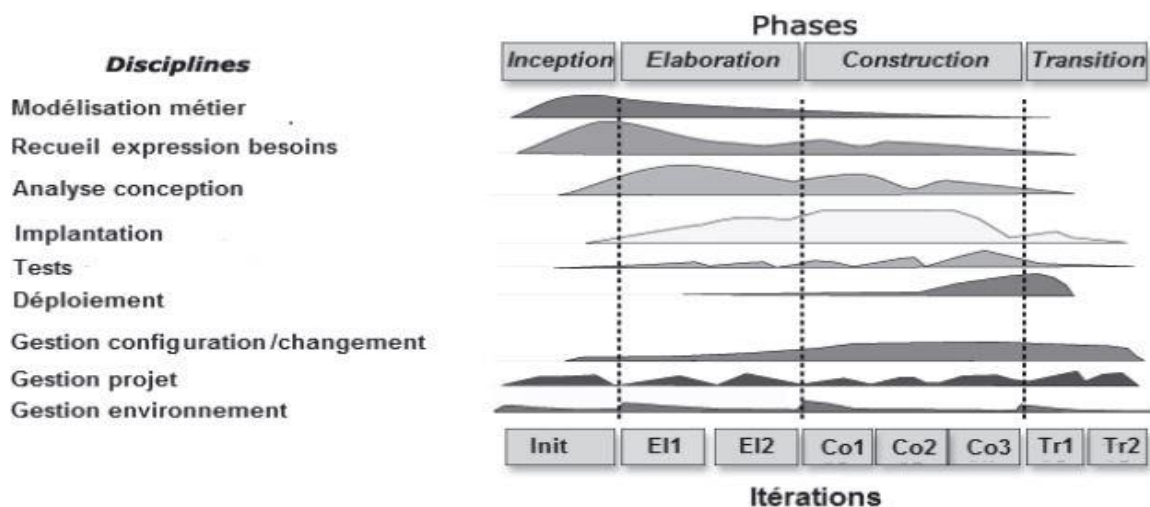
### **2.2.3. Les modèles agiles**

Les modèles agiles sont un mélange entre les modèles incrémentaux et itératifs visant à réduire le cycle de développement logiciel [Yannick, 2007]. Ils considèrent que les changements (des besoins des utilisateurs, mais également de l'architecture, de la conception, de la technologie) sont inévitables et doivent être pris en compte par le modèle de développement [Oriat, 2007]. Les modèles agiles cherchent à alléger le cadre (« *trop de méthode tue la méthode* ») et à rester le plus possible focalisées sur le code. Ils visent la simplicité, la légèreté, l'auto-adaptation et l'auto-organisation [Lonchamp, 2015]. Parmi les modèles agiles, nous présenterons les modèles RUP, XP et Scrum.

### 2.2.3.1. Le modèle RUP

RUP (Rational Unified Process) est un modèle de développement logiciel orientée objet apparue en 1998 [Lonchamp, 2015]. RUP définit un processus de développement relativement « *générique* », qui peut être adapté à des contextes particuliers [Lonchamp, 2015]. Dans cette approche, le développement d'un logiciel passe par quatre (04) phases, chacune pouvant donner lieu à une série d'itérations [Lonchamp, 2015]. Il s'agit de :

- **Le lancement (inception)** : c'est la phase la plus courte, elle se fait généralement en une itération. Elle explicite la vision associée au projet en termes de faisabilité, de risques et de périmètre du projet [Lonchamp, 2015]
- **L'élaboration** : cette phase est constituée de quelques itération. Elle correspond à l'identification, la stabilisation de la plupart des besoins, la spécification de la plupart des cas d'utilisation, la conception de l'architecture de référence, la programmation et le test des éléments d'architecture les plus importants [Lonchamp, 2015].
- **La construction** : c'est la phase la plus longue (plus de 50 % du cycle de développement), elle se fait par incrément. A chaque incrément, le produit doit contenir tout ce qui avait été planifié ainsi que le manuel d'utilisation associé. Par contre, il peut éventuellement rester quelques erreurs non encore traitées [Lonchamp, 2015].
- **La transition** : dans cette phase, une version du produit est livrée ainsi que les manuels d'utilisation. Les erreurs sont corrigés et les utilisateurs sont formés [Lonchamp, 2015].



**Figure 2.6 :** Phases, itérations et disciplines du modèle RUP [Lonchamp, 2015]

RUP doit être instancié et adapté en fonction des spécificités du projet, de sa complexité. Il présente l'avantage d'être très flexible. Cependant il reste un modèle nettement basée sur les processus et la documentation UML, ce qui a tendance à l'alourdir [Lonchamp, 2015].

### 2.2.3.2. Le modèle XP

Le modèle XP (Extreming Programming) a été mis sur pieds en 1999 par Ward Cunningham et Kent Beck comme un cadre technique pour le développement agile des applications. Il est conçu pour pousser « à l'extrême » les meilleures pratiques du développement logiciel tout en intégrant le client à l'équipe de projet [Lonchamp, 2015]. Le modèle XP renvoie à un ensemble de principes et de pratiques allant de la programmation à la collaboration en passant par l'organisation des équipes. Ces principes et pratiques sont présentés dans la figure suivantes.

Gestion de projet	-Planification itérative (cycle de planification hebdomadaire, réunion hebdomadaire ou « planning game » et réunion quotidienne ou « stand-up meeting ») -La livraison itérative et incrémentale.
Programmation	-La conception simple, incrémentale -Les tests unitaires et fonctionnels -La construction rapide -Le déploiement quotidien -Le remaniement des codes -L'intégration continue
Collaboration	-La métaphore -La programmation en paire -L'appropriation collective du code -Les règles de codage -Le client sur le site

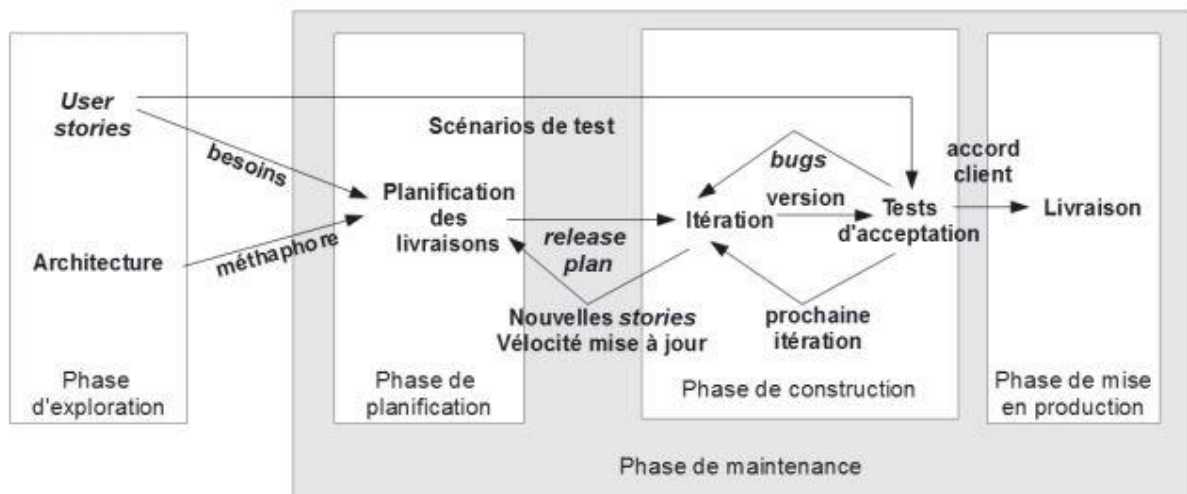
**Figure 2.7 :** Principes et pratique d'XP [Khalil, 2011]

Le cycle de vie de développement du modèle XP est composé de cinq (05) phases :

- **Une phase d'exploration** pendant laquelle les *user stories* initiales et les éléments architecturaux initiaux du projet (concepts et composants) sont déterminés avec les clients [Lonchamp, 2015]. Une user story est une phrase en langage naturel, qui vise à décrire avec clarté un besoin, c'est-à-dire une fonctionnalité à développer, une qualité espérée ou une contrainte à respecter par l'application [Lonchamp, 2015].

- **Une phase de planification** pendant laquelle sont sélectionnées avec les clients les *user stories* à implanter dans la première livraison et les livraisons suivantes (*release plan*). Les *user stories* choisies pour la première livraison sont décomposées en tâches à réaliser dont les durées sont estimées par les développeurs [Lonchamp, 2015].
- **Une phase de construction incrémentale de la livraison** pendant laquelle les itérations d'une durée d'une à quatre semaines sont planifiées de manière souple [Lonchamp, 2015]. Les binômes sont créés, puis les tests sont codés. [Yannick, 2007]. Une fois l'ensemble des tests fonctionnels (d'acceptation) passés, on entame la mise en production de la livraison
- **Une phase de mise en production de la livraison** impliquant l'accord du client [Lonchamp, 2015].
- **Une phase de maintenance** qui répète les phases de planification, construction et mise en production pour les livraisons suivantes [Lonchamp, 2015].

Ce cycle se répète tant que le client peut sélectionner des *stories* à livrer.



**Figure 2.8** : Phases du modèle XP [Lonchamp, 2015]

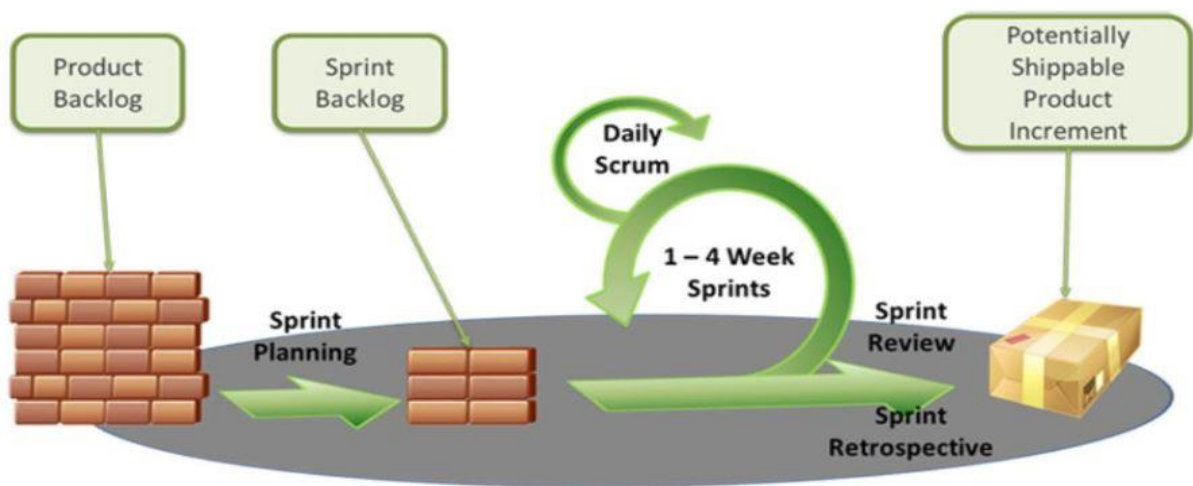
La production de logiciels de bonne qualité bien adaptés aux besoins des clients et l'efficacité pour les petits projets sont les points le plus souvent portés à l'actif de XP [Lonchamp, 2015]. Cependant il impose au client d'être en plein temps sur le site de développement de l'application.



### 2.2.3.3. Le modèle Scrum

Scrum est un cadre de gestion d'un projet agile et signifie « mêlée de rugby ». L'image est qu'on fait progresser le ballon en travaillant ensemble, comme on fait progresser un projet en travaillant ensemble [Lonchamp, 2015]. Ce modèle qualifie un ensemble de rôles, d'instruments de gestion et de pratiques managériales favorisant un environnement basée sur les piliers tels que la transparence, l'inspection, le suivi et l'adaptation [Khalil, 2011]. Le cycle de vie d'un projet Scrum comprend trois phases :

- **La Phase initiale** : cette phase a pour but de préparer le travail à faire. Elle doit construire la liste des éléments que comportera la version du produit. Elle comporte les objectifs suivants : Planning -Architecture, Mise en place d'un *backlog produit* (liste des éléments à effectuer), Définition de l'équipe, Analyse des risques [TREMBLAY, 2007].
- **Les phases de sprints** : Un *sprint backlog* (liste des tâches à réaliser) est sélectionné.  
Cette liste sera suivie et réalisée sur une période d'environ 7 à 30 jours, au cours de laquelle l'équipe est isolée des influences extérieures. À la fin de cette phase, une itération exploitable du produit est fournie [TREMBLAY, 2007].
- **La Phase de clôture** : Elle prépare la documentation, finalise les tests et rend la version fonctionnelle et présentable [TREMBLAY, 2007].



**Figure 2.9 :** Modèle Scrum [Vickoff, 2009]

Ce modèle a l'avantage qu'il s'adapte à tout projet de développement logiciel et accorde une très grande importance au partage des connaissances au sein de l'équipe

grâce aux réunions journalières. Sa principale faiblesse est qu'il ne prend pas en compte des modifications urgentes au cours du Sprint.

Par rapport aux contraintes auxquelles est soumis un travail de développement de logiciel, certains modèles peuvent être plus adaptées que d'autres [Yannick, 2007].

Le tableau ci-dessous donne un comparatif des différents modèles suivant certains critères.

<b>Critères</b>	<b>Modèles Linéaires</b>	<b>Modèles itératifs et incrémentaux</b>	<b>Modèles agiles</b>
Objectif	Respecter le besoin initial et les engagements	Satisfaire l'utilisateur	Satisfaire l'utilisateur
Changement	Opposé au changement ou, en tout cas, moins enclin à l'accepter	Accepter le changement	Accepter le changement
Livraison	Livrer en une seule fois une application « finalisée »	Livrer fréquemment et change en fonction des besoins client	Livrer fréquemment
Communication	Communication avec le client au début et à la fin	Grande communication avec le client	Grande communication avec le client
Équipe	Travailler en synergie	Travailler en synergie	Travailler de façon segmentée
Difficulté	Non stabilité des besoins	explosion de besoins dans les Projets de grande envergure	Délais extrêmement courts

**Tableau 2.2 :** Tableau comparatif des modèles de développement

Les modèles agiles semblent être les mieux adaptés pour le développement de notre solution.

## 2.3. Pratiques Pédagogiques

Dans la partie qui précède nous avons présenté différents modèles de développement logiciel. Mais le volet pédagogique n'y est pas pris en compte. Or selon



[Barette, 2009] « *les TIC ne se révèlent efficaces que lorsqu'elles s'intègrent à une pédagogie qui articule finement des activités pédagogiques dont les méthodes servent des objectifs explicites* ». L'outil que nous voulons réaliser étant un outil pédagogique, nous présenterons dans cette partie quelques pratiques pédagogiques sur lesquelles s'appuyer pour développer un tel outil.

### **2.3.1. Théories de l'apprentissage**

Les théories de l'apprentissage sont un ensemble organisé de principes expliquant la manière dont les individus acquièrent, conservent et se remémorent les connaissances au cours de l'expérience d'apprentissage. L'application directe d'une théorie de l'apprentissage permet de formuler des hypothèses de travail et des méthodes pour des recherches en didactique plus systématique [Chekour et al., 2015]. Il est donc important pour l'enseignant de savoir quelle théorie d'apprentissage utiliser lors du processus d'enseignement pour faciliter au mieux le processus d'apprentissage chez l'apprenant. Nous présenterons ici les quatre (04) grandes théories de l'apprentissage.

#### **2.3.1.1. Behaviorisme**

Né de Pavlov et Watson, le behaviorisme est une théorie de l'apprentissage qui s'intéresse à l'étude des comportements observables sans faire appel à des mécanismes internes au cerveau ou des processus mentaux non directement observables [Good & Brophy, 1990]. Le behaviorisme se concentre sur la façon dont les élèves apprennent. Il se focalise sur l'idée que tous les comportements sont appris par l'interaction avec l'environnement. L'acquisition des connaissances se fait par paliers successifs et le passage d'un niveau de connaissance à un autre s'opère par des renforcements positifs des réponses et comportements attendus [Chekour et al., 2015].

Cette théorie d'apprentissage est essentielle pour les enseignants car ils influent sur la façon dont les élèves réagissent et se comportent. L'enseignant répète une notion plusieurs fois lorsqu'il constate à travers le comportement auprès des apprenants que celle-ci n'est pas assimilée et conçoit les exercices progressifs pour guider les élèves dans leurs réalisations [Chekour et al., 2015]. Les élèves participent de manière passif à l'apprentissage comportementale.

### 2.3.1.2. Cognitivism

Le Cognitivism naît en 1958 en même temps que l'intelligence artificielle. Contrairement au béhaviorisme, le cognitivism se concentre sur les processus impliqués dans l'apprentissage plutôt que sur les comportements observés [Chekour et al., 2015]. Le cognitivism considère trois catégories de connaissances définies par [Tardif, 1992] comme suit :

- **Les connaissances déclaratives** : les faits, les règles, les lois, etc. Elles répondent à la question du « quoi ? ».
- **Les connaissances procédurales** : les procédures à suivre, les étapes de réalisation, etc. Elles définissent le « comment ? ».
- **Les connaissances conditionnelles** : les moments et contextes de réalisation d'une procédure. Elles renvoient au « quand ? ».

Les apprenants sont des participants actifs dans le processus d'apprentissage. Ils utilisent diverses stratégies pour traiter et construire leur compréhension personnelle du contenu. L'enseignant quant à lui est un gestionnaire des apprentissages : il guide, anime, dirige, conseille, explique et remédie [Chekour et al., 2015].

### 2.3.1.3. Constructivism

Le constructivism est né en 1975 des travaux de Piaget. Cette théorie est fondée sur l'idée selon laquelle les apprenants construisent activement leur propre savoir sur la base de leurs expériences. Les apprenants utilisent essentiellement leurs connaissances comme base et s'en servent avec de nouvelles choses qu'ils apprennent. Les expériences individuelles de chacun rendent leur apprentissage unique.

Le constructivism s'appuie sur deux principes : l'assimilation et l'accommodation qui forment un couple indispensable à l'activité cognitive d'équilibration [Piaget, 1975]. Alors que l'assimilation désigne la réintégration d'éléments externes nouveaux dans une structure interne préexistante, l'accommodation désigne l'adaptation de l'organisme aux variations externes qu'il ne réussit pas à assimiler [Piaget, 1975].

Dans cette théorie, L'élève est auteur de son propre savoir, il construit ses connaissances par assimilation et accommodation. L'enseignant quant à lui est chargé de mettre en place le cadre adéquat au processus d'apprentissage.

### 2.3.1.3. Sociocconstructivisme

Né de Vygotsky, le sociocconstructivisme reprend les idées principales du constructivisme en y ajoutant le rôle social des apprentissages [Chekour et al., 2015]. Le sociocconstructivisme voit l'apprentissage comme l'acquisition de connaissances grâce aux échanges entre enseignant et élève ou entre élèves. « *Les élèves n'apprennent pas seulement grâce à la transmission des connaissances par l'enseignant mais aussi grâce aux interactions* » [Doise & Mugny, 1981]. Le sociocconstructivisme repose sur deux principes :

- **La Zone Proximale de Développement (ZPD)** : qui représente l'écart entre ce que l'apprenant peut faire seul et ce qu'il peut faire avec l'aide de quelqu'un plus compétent.
- **Le More Knowledgeable Other (MKO)** : qui se réfère à toute personne qui a un niveau de capacité plus grand que l'apprenant ou une compréhension meilleure à l'égard d'une tâche particulière. L'enseignant est généralement considéré comme le MKO.

Les apprentissages doivent être situés dans la ZPD (Zone Proximale de Développement) de l'apprenant car cette zone augmente nettement le potentiel de l'élève à apprendre plus efficacement [Vygotsky, 1980].

Le travail est collaboratif entre élèves et entre élèves et enseignant. L'enseignant crée des situations-problèmes, pour permettre aux apprenants de résoudre de vrai problème. Il les confronte à des documents authentiques tels que des articles, des sites internet etc. Plutôt qu'à des documents didactisés.

Le tableau suivant présente un récapitulatif des différentes théories précitées.

	<b>Behaviorisme</b>	<b>Cognitivisme</b>	<b>Constructivisme</b>	<b>Socioconstructivisme</b>
Postulat de base	L'apprentissage est la modification durable du comportement observable	L'apprentissage est une succession d'étapes, résultant de la modification des processus mentaux	La connaissance est construite par l'apprenant sur la base de ses expériences	La connaissance est construite par l'apprenant sur la base des interactions.
Principe	Conditionnement, renforcement	Intégration des schémas mentaux	Assimilation et Accommodation	ZPD, MKO
Méthodes	L'exposé, la pratique répétée et le renforcement	Un enseignement individualisé, interactif et stratégique	Manipulations et expérimentation	Apprentissage collaboratif
Rôle de l'enseignant	Transmetteur d'informations	Facilitateur	Guide	Médiateur
Rôle de l'apprenant	Passif	Actif	Auteur	Pair

**Tableau 2.3 :** récapitulatif des théories de l'apprentissage

### 2.3.2. Approches pédagogiques

Selon [Meirieu, 1991] la pédagogie est une « *réflexion sur l'éducation de l'enfant (...) la pédagogie s'interroge sur les finalités (...), sur la nature des connaissances à transmettre et sur la méthode qu'elle doit utiliser* ». Les théories d'apprentissage ont donné naissance aux approches pédagogiques. Il en existe plusieurs, nous en présenterons trois (03).

#### 2.3.2.1. La pédagogie par objectifs

La pédagogie par objectif (PPO) est issue de la théorie behavioriste. Cette approche se base sur le fait qu'un enseignement est mieux assimilé s'il repose sur des objectifs bien définis. Les objectifs pédagogiques sont vus comme des outils de travail qui orientent l'activité d'enseignement et l'activité d'apprentissage [Feumo, 2019]. Il existe cinq (05) niveaux de définition des objectifs parmi lesquels trois (03) qui se rapportent à un cours :

- **Objectif général :** il se situe au niveau de la planification du cours.

- **Objectif spécifique** : il se situe au niveau de la planification d'une leçon et dit ce que l'élève sera capable de faire à la fin.
- **Objectif opérationnel** : il décrit ce qui est attendu dans le comportement de l'apprenant pendant et après une leçon.

D'après [Daniel, 1988] pour qu'un objectif spécifique devienne opérationnel, il doit respecter les exigences suivantes :

- Il doit être énoncé de la manière la moins équivoque possible.
- Il doit décrire une activité de l'apprenant identifiable par un comportement observable.
- Il doit mentionner les conditions dans lesquelles le comportement escompté doit se manifester.
- Il doit indiquer le niveau d'exigence auquel l'apprentissage est tenu de se situer, et les critères qui serviront à l'évaluation de cet apprentissage.

Les avantages de cette approche sont qu'elle permet aux enseignants de confectionner des programmes d'enseignements, de préparer des activités de façon spécifique et détaillé. Une fois les objectifs d'un enseignement énoncés, les apprenants savent ce qui est attendu d'eux et peuvent mener des apprentissages individualisés. Cependant elle ajoute une grande complexité à la planification des enseignements et ignore les apprentissages dont les résultats sont peu prévisibles [Feumo, 2019].

### 2.3.2.2. La pédagogie par projet

La pédagogie par projet est issue de la théorie socioconstructiviste. Comme son nom l'indique, la pédagogie par projet a pour objectif de faire passer à un collectif d'apprenants, des apprentissages à travers la réalisation d'une production concrète (résultat concret), socialisable, et qui intègre des savoirs nouveaux [Dumont, 2017]. L'enseignant a un rôle d'accompagnateur, il apporte des outils, discute avec les apprenants, aide à la régulation des groupes et assure l'aboutissement du projet [Dumont, 2017].

L'avantage de cette pédagogie est qu'elle favorise l'acquisition des connaissances pratiques chez l'apprenant. Toutefois, dans les groupes, certains apprenants laissent toutes les tâches aux autres.

### 2.3.2.3. L'approche par compétences

L'approche par compétence (APC) est issue des théories behavioriste et constructiviste. C'est une approche pédagogique qui vise à développer chez l'apprenant des **compétences** durables susceptibles de l'aider dans son parcours éducatif et dans la vie quotidienne [Ghazel, 2012]. Une compétence est vue par [Perrenoud, 2000] comme *« une capacité d'action efficace face à une famille de situation, qu'on arrive à maîtriser parce qu'on dispose à la fois des connaissances nécessaires et de la capacité de les mobiliser à bon escient, en temps opportun, pour identifier et résoudre de vrai problèmes »*.

Comparé aux approches traditionnelles qui définissent et découpent une formation en périodes temporelles et thématiques, dans l'APC une formation est définie et découpée en termes d'acquisition de capacités nécessaire pour effectuer une tâche [Ghazel, 2012]. C'est l'application d'une capacité dans un contexte particulier avec un niveau de maîtrise spécifié qui définit la compétence visée [Ghazel, 2012].

Le point majeur de cette approche est qu'elle place l'apprenant au centre de l'action éducative, il est le principal acteur dans le processus enseignement apprentissage. On peut dire qu'elle répond parfaitement au besoin de réduire l'échec scolaire [Ghazel, 2012].

Le tableau suivant donne un récapitulatif des différentes approches pédagogiques.

	<b>Pédagogie par objectif</b>	<b>Pédagogie par Projet</b>	<b>Approche par compétences</b>
Théories	Behaviorisme	Socioconstructivisme	Behaviorisme et constructivisme
Enseignement-apprentissage	S'articule autour des objectif	S'articule autour des projet	S'articule autour des compétences
Rôle de l'enseignant	Définir les objectifs à atteindre par les apprenants	Organiser les groupes, poser des situations problèmes aux apprenants	Définir les compétences à développer chez l'apprenant
Rôle de l'apprenant	Suivre l'apprentissage et atteindre les objectif	Travailler à résoudre la situation problème	Développer les compétences
Avantages	La formation est structurée, les activités sont détaillées	Favorise l'acquisition des connaissances pratiques	Réduit fortement l'échec scolaire

Inconvénients	Ignore les apprentissages dont les résultats ne sont pas prédéfinis	Certains apprenants ne travaillent pas dans les groupes	
---------------	---	---	--

**Tableau 2.4 : Récapitulatif des approches pédagogiques**

## 2.4. Conception ergonomique des logiciels

Nous avons précédemment présenté les modèles de développement logiciel qui vont nous permettre de produire un logiciel de bonne qualité et les pratiques pédagogiques qui feront de ce logiciel un logiciel pédagogique. Pour que notre outil soit accepté par les utilisateurs (enseignants, élèves, parents) il faut qu'il prenne en compte leurs caractéristiques physiologique, perceptives et cognitive : il faut qu'il soit ergonomique.

[BRANGIER, 1990] définit l'ergonomie des logiciels comme la compatibilité des logiciels avec les caractéristiques physiologiques, perceptives et cognitives des utilisateurs. Son objectif est d'améliorer l'interface des hommes avec les logiciels en tenant compte de l'utilité et de l'utilisabilité de ces logiciels. Nous présenterons les critères d'utilité et d'utilisabilité des logiciels (critères ergonomiques), et les méthodes d'évaluation de ces derniers.

### 2.4.1. Utilité et utilisabilité des logiciels

Le logiciel doit servir à un besoin (utilité), et le logiciel doit être utilisé de façon efficace (utilisabilité). L'**utilité** renvoie à l'idée qu'un logiciel doit répondre à un besoin et être pertinent au regard des objectifs de l'utilisateur. L'**utilisabilité** correspond au « *degré selon lequel un logiciel peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié* » [ISO 9241-11, 1998].

Afin de mesurer l'adéquation entre l'utilité et l'utilisabilité d'un logiciel, [Bastien & Scapin, 1993] ont défini dix-huit (18) critères repartis suivants huit (08) dimensions :

- ❖ **Guidage** : ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec le logiciel.

- ❖ **Charge de travail** : optimiser les éléments de l'interface qui ont un rôle dans la réduction de la charge perceptive ou mnésique des utilisateurs, de même que dans l'augmentation de l'efficacité du dialogue.
- ❖ **Contrôle explicite** : C'est la prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leurs actions.
- ❖ **Adaptabilité** : capacité du système à réagir selon le contexte et selon les besoins et les préférences des utilisateurs.
- ❖ **Gestion des Erreurs** : Ce sont les moyens permettant d'une part d'éviter ou de réduire les erreurs, d'autre part de les corriger lorsqu'elles surviennent.
- ❖ **Homogénéité/Cohérence** : Les choix de conception d'interface doivent être conservés pour des contextes identiques, et doivent être différents pour des contextes différents.
- ❖ **Signifiante des Codes et Dénominations** : Il doit y avoir adéquation entre l'objet ou l'information affichée ou entrée, et son référent.
- ❖ **Compatibilité** : Il faut qu'il y ait accord entre les caractéristiques des utilisateurs et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part.

### 2.4.2. Méthodes d'évaluation ergonomique

L'évaluation ergonomique d'un logiciel permet de vérifier le respect des critères ergonomiques. A l'issue de celle-ci, il en ressort si le logiciel est adapté ou non à l'utilisateur final. Il existe plusieurs méthodes d'évaluation ergonomique des logiciels parmi lesquels deux (02) grands types se distinguent :

#### 2.4.2.1. Les Méthodes basées sur les utilisateurs

Ces méthodes font appel à l'expérience d'utilisation du logiciel par les utilisateurs. On distingue :

- **Le test utilisateur** : il sert à vérifier les critères d'utilité, d'utilisabilité et de satisfaction, en faisant tester l'interface par des utilisateurs individuellement.
- **Le focus groupe** : il permet de recueillir l'opinion des utilisateurs réels ou potentiels sur un logiciel ou un site Web.



#### 2.4.2.2. Les Méthodes basées sur l'expertise de l'ergonome

Ces méthodes font appel à l'expertise d'un ergonome pour évaluer le logiciel.

On distingue :

- **L'audit ergonomique** (ou inspection experte) : il consiste à vérifier si le logiciel respecte un ensemble de critères d'utilisabilité.
- **Le prototypage** : il sert à consolider le design de l'interface homme-machine. Bien que cette démarche se base sur des méthodes faisant appel aux utilisateurs, l'ergonome ou le designer réalise seul cette étape. Le prototypage est d'autant plus efficace lorsqu'il est itératif.

Tout au long de la réalisation de notre outil, nous nous attacherons à prendre en compte les besoins et attentes de l'utilisateur final afin d'intégrer la composante humaine, le « *point de vue utilisateur* », dans le processus de conception.

## 2.5. Pratiques de recherche

Tout travail de mémoire est constitué de recherches. Celles-ci définissent un ensemble de pratiques et de méthodes qui constituent la méthodologie de recherche. La méthodologie de recherche est donc définie par [Claude, 2019] comme l'ensemble des règles et des démarches adoptées par un chercheur pendant son travail de recherche pour parvenir à une ou plusieurs conclusions. Cette méthodologie fait référence à la population d'étude, aux méthodes de recherche, aux instruments de collecte de données et aux méthodes d'analyse de données. Nous présenterons ici les différentes pratiques de la recherche.

### 2.5.1. Population et échantillon d'étude

Avant tout travail de recherche, il est important pour le chercheur de cibler la population sur laquelle va s'effectuer l'étude.

#### 2.5.1.1. Population d'étude

La population d'étude est un ensemble fini d'objets (les individus) sur lesquels l'étude se porte et dont les éléments répondent à une ou plusieurs caractéristiques communes [Wikipédia, 2019]. Le choix de la population d'étude s'effectue suivant certains critères :

- Les caractéristiques de la population (âge, sexe, niveau d'étude)
- La localisation géographique de la population

Il arrive souvent que la population d'étude soit très grande et que les moyens du chercheur ne lui permettent pas de couvrir l'effectivité de la population. Dans ce cas il définit un échantillon sur lequel mener son étude.

#### 2.5.1.2. Échantillonnage

Un échantillon est un sous-ensemble d'individus d'une population large à partir duquel, un comportement de la population d'étude peut être généralisable [Claude, 2019]. L'échantillonnage est un moyen de sélectionner un sous-ensemble précis d'individus d'une population afin de réaliser une étude [Claude, 2019]. Il existe deux grandes méthodes d'échantillonnage :

- **L'échantillonnage probabiliste** : la sélection repose sur un principe de tirage aléatoire (les individus sont sélectionnés au hasard). Chaque individu a une chance d'être sélectionné.
- **L'échantillonnage non probabiliste** : la sélection est subjective c'est-à-dire non aléatoires. L'échantillon retenu à l'issue de la sélection est considéré comme représentatif car il possède les mêmes caractéristiques que la population étudiée.

#### 2.5.2. Les méthodes de recherche

Une fois la population déterminée, le chercheur choisit la méthode de recherche à appliquer en fonction des caractéristiques de la population et du type de résultat attendu. Il existe principalement deux méthodes de recherche à savoir la méthode quantitative et la méthode qualitative.

##### 2.5.2.1. La méthode quantitative

La méthode de recherche quantitative sert à prouver ou démontrer des faits en quantifiant un phénomène [Claude, 2019]. Elle utilise des outils d'analyse mathématiques et statistiques en vue de décrire, expliquer et prédire le phénomène étudié. Cette méthode s'appuie sur une collecte de données obtenues grâce à deux techniques principales :

- **Le sondage** : qui consiste à poser *une question* à la population étudiée et à recueillir les réponses [Claude, 2019].

- **Le questionnaire** : qui consiste à soumettre la population étudiée à une série de questions [Claude, 2019].

Les résultats d'une étude quantitative s'expriment en données chiffrées sont présentés sous forme de graphiques ou de tableaux statistiques.

#### 2.5.2.2. La méthode qualitative

La recherche qualitative est plus descriptive et se concentre sur des interprétations, des expériences et leur signification [Claude, 2019]. Elle a pour but de comprendre ou d'expliquer un phénomène. De ce fait, elle laisse délibérément de côté l'aspect quantitatif pour gagner en profondeur dans l'analyse de l'objet d'étude. Cette méthode s'appuie sur une collecte de données obtenues grâce à deux techniques principales :

- **L'observation** : Elle consiste à s'immiscer dans l'environnement du sujet, et l'observer pendant une certaine période, permettant ainsi une analyse du réel [Claude, 2019].
- **L'entretien** : Elle consiste à une discussion verbale avec le sujet afin de recueillir son avis [Claude, 2019].

Les résultats d'une étude qualitative sont généralement exprimés avec des mots.

#### 2.5.3. Analyse des données

Une fois les données collectées auprès de la population, il faut les analyser selon l'objectif de la recherche afin d'établir les conclusions. Il existe plusieurs approches possibles. Le but étant de mettre en valeur les résultats pour en tirer un maximum d'information. On distingue principalement :

- **L'analyse de contenus** : c'est une méthode d'analyse appliquée au résultats obtenus à l'issue d'une étude qualitative. Elle permettra de ressortir les contenus de discours des sujets. Il s'agit alors de passer au crible de l'analyse en profondeur de termes utilisés, leur fréquence et leur mode d'agencement, la construction du discours et son développement [Wikipédia, 2020].
- **L'analyse quantitative** : c'est une méthode permettant d'analyser des données standardisées (résultant d'une enquête par questionnaire ou par sondage). L'analyse quantitative produit des informations chiffrées (pourcentages, effectifs, ratios...) représentées sous forme de tableaux ou de graphiques [Martin, 2019].

Dans un travail de recherche, les études qualitatives et quantitatives représentent des approches complémentaires.

Ainsi s'achève ce chapitre dans lequel nous avons présenté plusieurs méthodes, approches et pratiques destinées à nous guider dans la conception de l'outil pédagogique que nous envisageons réaliser. Dans le chapitre suivant, nous présenterons notre méthodologie de travail.

# Chapitre 3

## Matériel Et Méthodes

Ce chapitre sera constitué de trois (03) parties. La première présentera les méthodes et approches que nous utiliserons pour réaliser notre plateforme d'enseignement apprentissage nommée **Axiom**. La deuxième partie déroulera la méthodologie que nous allons suivre pour ce fait. La troisième partie sera dédiée à la présentation du matériel que nous allons utiliser pour le développement de la plateforme.

### 3.1. Choix du cadre conceptuel

#### 3.1.1. Choix du modèle de développement logiciel

Pour la réalisation de la plateforme Axiom, nous choisissons parmi les modèle agiles, le modèle **XP**. Notre choix est basé sur les raisons suivantes :

- XP est centré sur les pratiques de programmation : la conception est réalisée de façon simple et le code est remanié en permanence pour rester aussi clair et simple que possible.
- L'agilité de XP permet de prendre en compte les différents besoins du client et leur potentielle évolution dans un processus itératif.
- XP permet de contrôler la qualité du logiciel par l'implémentation des tests.
- XP est adaptée et facile à mettre en œuvre pour les petits projets, il accroît la productivité, la dynamique et la motivation des équipes.

### **3.1.2. Choix des pratiques pédagogiques**

La plateforme pédagogique Axiom devra permettre de créer un environnement d'apprentissage en ligne favorisant le développement des compétences chez les apprenants à travers les échanges et les interactions entre les apprenants, les enseignants et les contenus pédagogiques. Les enseignements dans Axiom seront construits suivant l'approche APC, qui est l'approche pédagogique en vigueur au Cameroun. Ce qui implique que les processus d'acquisition de connaissance seront considérés selon les approches behavioristes, constructivistes et socioconstructivistes.

### **3.1.3. Choix de la conception ergonomique**

Les interfaces de Axiom devront respecter les critères ergonomiques de Bastien et Scapin. L'évaluation de la plateforme se fera par des test utilisateurs. Après les tests, nous recueillerons les avis des utilisateurs sur la plateforme.

### **3.1.4. Choix de la méthode de recherche**

Dans le cadre de notre étude, nous avons opté pour la combinaison des méthodes de recherche quantitative et qualitative.

## **3.2. Méthodologie de travail**

### **3.2.1. Méthodologie de recherche**

#### **3.2.1.1. Population et échantillon d'étude**

Notre population cible est constituée des enseignants, des élèves et des parents d'élèves de l'enseignement secondaire du Cameroun. Ce pendant compte tenu du temps imparti pour notre recherche et de nos moyens limités, nous avons opté pour les enseignants, les élèves et les parents d'élèves du Lycée de Biyem-Assi, établissement dans lequel nous avons effectué notre stage académique.

Ne pouvant pas interroger l'ensemble des individus du Lycée de Biyem-Assi, nous avons sélectionné un échantillon sur lequel nous avons mené notre étude.

Pour sélectionner les élèves, nous avons utilisé l'échantillonnage probabiliste plus précisément l'échantillonnage aléatoire simple. Le but étant d'avoir également dans notre échantillon des élèves ayant déjà eu une expérience avec l'enseignement en ligne. Nous

avons donc fait un tirage au sort d'une part entre les classes qui suivent les classes d'examen (pour avoir des élèves ayant fait les cours en ligne en 2020) et d'autre part entre les classes restantes. A l'issu de ce tirage nous avons obtenu les élèves des classes suivantes (par la même occasion leurs parents) :

Classes	Nombre d'élèves
Terminale TI	43
Seconde C 2	35
Troisième Allemand 2	42
Cinquième 3	44
<b>Total</b>	<b>164</b>

**Tableau 3.1 : Classes sélectionnées pour l'étude**

L'échantillon des parents était constitué des parents des élèves sélectionnés. Nous avons pu mener notre étude sur **71 parents d'élèves**.

Pour sélectionner l'échantillon des enseignants, nous avons utilisé l'échantillonnage par choix raisonné ou par convenance qui est une variante de l'échantillonnage non probabiliste. Il consiste à opérer un choix sur les sujets à interroger en tenant compte des caractéristiques de leur profil, et de leur importance dans le phénomène étudié. Notre objectif était d'avoir dans notre échantillon une représentativité de toutes les disciplines. Ainsi nous avons eu :

Disciplines	Nombre d'enseignants
Allemand	2
Anglais	2
Espagnol	2
Français	2
Histoire-Géographie	2
Informatique	2
Mathématique	2
Physique-Chimie	2
Science	2
<b>Total</b>	<b>18</b>

**Tableau 3.2 : Enseignants sélectionnés**

Nous avons obtenu un échantillon d'une taille de 18 enseignants, 164 élèves et 71 parents d'élèves.

### **3.2.1.2. Collecte des données**

Pour récolter les données, nous avons utilisé les questionnaires pour les élèves et leurs parents, et l'entretien pour les enseignants.

- Nos questionnaires ont été construits clairement et sans ambiguïté. Certaines questions ont été formulées selon l'échelle de Likert. Le questionnaire a été soumis aux élèves qui l'ont rempli en présentiel. Puis à chacun d'eux nous avons remis un questionnaire à faire remplir par un des parents et à ramener le lendemain. Nos questionnaires étaient structurés comme suit :
  - Une introduction destinée à préciser l'objet de l'enquête.
  - Une deuxième partie destinée à recueillir l'identité du sujet.
  - Une troisième partie permettant d'appréhender : l'état d'utilisation des TIC dans l'apprentissage (pour élèves) ; le degré de suivi des élèves (pour les parents).
  - Une dernière partie permettant de relever les attentes quant à l'outil à développer.
- Pour l'entretien, nous avons formulé une série de questions et les avons posées aux enseignants. Le guide d'entretien comprenait :
  - Une introduction destinée à préciser l'objet de l'enquête.
  - Une deuxième partie destinée à recueillir l'identité du sujet.
  - Une troisième partie contenant les questions relatives à notre étude.

### **3.2.1.3. Analyses des données**

Nous avons fait une analyse quantitative sur les données recueillies avec les questionnaires et une analyse de contenu sur celles des entretiens.

- **Analyse quantitative :** Les données récoltées lors de l'enquête avec les questionnaires ont été analysées à l'aide du logiciel Excel. Nous avons dans un premier temps constitué une bibliothèque des variables relatives aux différentes questions. Par la suite, nous avons dépouillé chaque questionnaire et introduit les données dans le fichier Excel. Pour analyser et traiter ces données, nous avons utilisé des graphes en forme de secteurs.



- **Analyse de contenu :** Il s'agit pour nous de faire ressortir les contenus de discours des enseignants interrogés. En plus du contenu du discours des enseignants, nous avons également tenu compte des éléments de la communication non verbale qui nous ont permis de déceler les appréhensions de ces derniers face à l'utilisation des outils TIC.

### 3.2.2. Plan de réalisation de Axiom

La réalisation de Axiom va se dérouler suivant les cinq (05) phases du modèle XP.

#### 3.2.2.1. Exploration

La première phase va consister à ressortir les utilisateurs et à formuler les user stories initiales. A partir de ceux-ci nous allons définir l'architecture global de Axiom.

Activités	Description	Résultats
Identification des utilisateurs	Recenser les différents types d'utilisateurs de Axiom	Liste des types d'utilisateurs de Axiom
Elaboration des user stories	Recenser les différents besoins des utilisateurs	Liste des user stories
Elaboration de l'architecture global	Définir les composants de de l'application	Diagrammes et description textuelle de Axiom

**Tableau 3.3 :** Activités de la phase d'exploration

#### 3.2.2.2. Planification

Une fois les user stories et l'architecture de Axiom connues, la deuxième phase va consister à planifier la réalisation des différentes user stories en attribuant à chacune d'elle un niveau de priorité et une durée.

Activités	Description	Résultats
Attribution des priorités aux user stories	Sélectionner et attribuer les priorités et les durées aux user stories	Liste de user stories avec priorité
Planification des itérations	Planifier les itérations en fonction des priorités de user stories.	Planning des itérations

**Tableau 3.4 :** Activités de la phase de planification

### 3.2.2.3. Construction incrémentale

Pendant cette phase nous allons développer au cours d'une itération les différentes user stories sélectionnées. Pour cela nous allons définir l'architecture de l'itération et les différentes tâches à exécuter.

Activités	Description	Résultats
Elaboration de l'architecture de l'itération	Donner une description de l'itération et de ses composants	Diagrammes et description textuelle de l'itération
Décomposition de l'itération en tâches	Définir les différentes tâches à réaliser pendant l'itération	Plan de l'itération
Développement des user stories	Développer les user stories selon le planning	Fonctionnalités de Axiom codées

**Tableau 3.5 :** Activités de la phase de construction incrémentale

### 3.2.2.4. Mise en production

Au cours de cette phase nous allons effectuer un test d'acceptation sur le livrable obtenu lors de la phase de construction incrémentale, afin de valider ce dernier.

Activités	Description	Résultats
Exécution du test d'acceptation sur	Manipuler la plateforme pour s'assurer que les fonctionnalités codées répondent aux besoins	Résultat du test

**Tableau 3.6 :** Activités de la phase de mise en production

### 3.2.2.5. Maintenance

Il s'agira ici de la répétition des phases précédentes (planification, construction, mise en production) dans le but d'ajouter de nouvelles fonctionnalités à Axiom.

## 3.3. Matériel

Tout au long de la conception et du développement de Axiom, nous serons amenés à utiliser divers outils.

### 3.3.1. Les logiciels

Nous aurons besoin des logiciels :

**Logiciel de conception :**

- **StarUml** : pour la réalisation des diagrammes.

**Logiciels de développement :**

- **WAMPserver** : pour héberger notre plateforme en local.
- **Sublime Text 2** : pour éditer les codes sources.
- **Les navigateur Mozilla Firefox, Google Chrome, Microsoft Edge** : pour visualiser et manipuler l'application.
- **Photoshop CS6** : pour créer le logo et certaines images.

### 3.3.2. Les langages informatiques

Nous utiliserons les langages :

**Langage de modélisation :**

- **UML (*Unified modeling language*)** : pour concevoir et modéliser l'architecture.

**Langages de programmation :**

- **HTML (*Hyper Text Markup Language*)** : pour la réalisation des différentes vues (interfaces) de l'application.
- **CSS (*Cascading Style sheets*)** : pour la mise en forme des vues.
- **JavaScript** : pour la gestion des évènements.
- **PHP (*Hypertext Preprocessor*)** : pour dynamiser l'application.

**Frameworks :**

Les frameworks permettent de faciliter le travail de développement. Nous aurons à utiliser :

- **Bootstrap** : framework CSS utilisé pour la mise en forme.
- **JQuery** : framework JavaScript utilisé pour la gestion des évènements.
- **PHPUnit** : un framework PHP permettant d'effectuer des tests unitaires.

### 3.3.3. Le matériel

Comme matériel physique, nous utiliserons :

- **Un ordinateur portable Dell** contenant le système d'exploitation **Windows 10**
- **Un smartphone Android** : pour test le rendu de l'application sur les petits écrans.

### 3.3.4. Les documents

Afin de structurer le contenu de Axiom suivant l'approche APC, nous allons nous inspirer des documents suivants :

- **La fiche de progression** (projet pédagogique) : qui permet à l'enseignant de planifier l'évolution du cours sur l'année.
- **La fiche de préparation d'une leçon** : qui permet à l'enseignant de définir les activités d'une leçon.

# Bibliographie

ATutor. (2020). *ATutor*. <https://atutor.ca>

Barette, C. (2009). Réussir l'intégration pédagogique des tic. *Revue internationale des technologies en pédagogie universitaire*, 6(2-3), 18-25.

Bastien, J. M., & Scapin, D. (1993). *Critères Ergonomiques pour l'Evaluation d'Interfaces Utilisateur* (INRIA).

BRANGIER, E. (1990). *Ergonomie des logiciels :approche psycho-ergonomique de l'interaction homme-ordinateur*.

Chekour, M., Laafou, M., & Janati-Idrissi, R. (2015). *L'évolution des théories de l'apprentissage à l'ère du numérique*.

Claude, G. (2019). *Etude qualitative et quantitative : Définitions et différences*.

<https://www.scribbr.fr/methodologie/etude-qualitative-et-quantitative/>

COTINAT, O. (2017). *Etude de deux plate-formes de e-formation : ATutor & Blackboard*.

Daniel, H. (1988). *Les objectifs pédagogiques en formation initiale et en formation continue* (PUF, Paris).

De Vries, E. (2001). Les logiciels d'apprentissage : Panoplie ou éventail ? *Revue française de pédagogie*, 105-116.

DJOUMESSI, K. (2019). *CONCEPTION ET RÉALISATION D'UN OUTIL D'AIDE A L'APPRENTISSAGE EN SVTEEB SUR LES PRATIQUES POUR ÉVITER LA CONTAMINATION MICROBIENNE EN CLASSE DE TROISIÈME L'ENSEIGNEMENT SECONDAIRE GÉNÉRAL*. Université de Yaoundé 1 : Ecole normale supérieure.

DOGBE-SEMANOU, D., DURAND, A., LEPROUST, M., & VANDERSTICHEL, H. (2008). *Etude comparative de plates-formes de formation à distance*.

Doise, W., & Mugny, G. (1981). *Le développement social de l'intelligence* (FeniXX).

- Dumont, S. (2017). *Les différentes approches pédagogiques*.
- Fâtimah, A. (2020). *ANALYSE, CONCEPTION ET REALISATION D'UNE PLATEFORME GENERIQUE FACILITANT L'ENSEIGNEMENT-APPRENTISSAGE DANS L'ESG*. Université de Yaoundé 1 : Ecole normale supérieure.
- Feumo. (2019). Les approches pédagogiques. *le blog de feumo*.
- FOTCHA, E. (2008). *Didacticiel multimédia pour le génie électrique accessible sur la plateforme Claroline*. Université de Douala : Ecole Normale Supérieure d'Enseignement Technique.
- Gérard, P. (2008). *Processus de Développement Logiciel*.
- Ghazel, T. (2012). *L'approche par compétence : Définition et principes*.
- Good, T., & Brophy, J. (1990). *Educational psychology : A realistic approach*. Addison Wesley Longman.
- Google Classroom. (2020). *Google Classroom*. <https://classroom.google.com>
- ISO 9241-11. (1998). *Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) – Partie 11 : Lignes directrices relatives à l'utilisabilité*.  
[http://www.iso.org/iso/fr/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=16883](http://www.iso.org/iso/fr/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16883)
- Karsenti, T., Collin, S., and Harper-Merrett, T. (2011). *Intégration pédagogique des tic : Succès et défis de 87 écoles africaines*.
- Karsenti, T., Poellhuber, B., Roy, N., & Parent, S. (2020). Le numérique et l'enseignement au temps de la COVID-19 : Entre défis et perspectives—Partie 1. *Revue internationale des technologies en pédagogie universitaire*, 17(2), 1-4. <https://doi.org/10.18162/ritpu-2021-v18n1-01>
- Khalil, C. (2011). *Les méthodes « agiles » de management de projets informatiques : Une analyse « par la pratique »*. Paris.

- Les Nations Unies au Cameroun. (2020). *Impact du Covid-19 sur le système éducatif du Cameroun*. <https://cameroon.un.org/fr/40118-impact-du-covid-19-sur-le-systeme-educatif-du-cameroun>
- Lonchamp, J. (2015). *Analyse des besoins pour le développement logiciel*.
- Mafouen, Achile, & Kouakep, Y. (2020). *Impact de la Covid-19 sur l'éducation au Cameroun et la mise à contribution du numérique en anglais (Nord—Centre—Littoral—Ouest) et mathématiques (Adamaoua – Nord) Le cas des groupes whatsapp*.  
<http://www.adjectif.net/spip/spip.php?article543>
- MAGNE NKUATE, B. (2018). *DEVELOPPEMENT D'UN OUTIL D'AIDE A L'APPRENTISSAGE D'UN COURS DE SECOURISME ET ACCIDENTS DE L'APPAREIL MOTEUR EN CLASSE DE 4emeESG*. Université de Yaoundé 1 : Ecole normale supérieure.
- Martin, O. (2019). Analyse Quantitative. *Presses Universitaires de France*, 10(2), 26.
- Meirieu, P. (1991). *Le Choix d'éduquer : Éthique et pédagogie*.
- Moodle. (2021). *Moodle*. <https://moodle.net>
- Mukherjee, M. (2013). *Technological tools for science classrooms : Choosing and using for productive and sustainable teaching and learning experiences* [PhD thesis]. The University of Queensland.
- Ntap, E. J. (2020). *Le Covid-19 révèle au grand jour le fossé numérique dans le système éducatif camerounais*. VOA. <https://www.voaafrique.com/a/le-système-éducatif-impacté-par-le-covid-19-au-cameroun/5363595.html>
- Oriat, C. (2007). *Analyse, Conception et Validation de Logiciels*.
- Perrenoud, P. (2000). *L'approche par compétence, une réponse à l'échec scolaire ?*
- Piaget, J. (1975). *L'équilibration des structures cognitives* (PUF).
- Rota, V. (2009). *Gestion de projet : Vers des méthodes agiles* (Eyrolles).
- Szilas, N., & Venni, J. (2020). *Google Classroom*.  
[https://edutechwiki.unige.ch/fr/Google\\_Classroom](https://edutechwiki.unige.ch/fr/Google_Classroom)

- Tardif, J. (1992). *Pour un enseignement stratégique : L'apport de la psychologie cognitive* (Editions Logiques).
- TREMBLAY, R. (2007). *IMPLANTATION D'UNE MÉTHODE AGILE DE DÉVELOPPEMENT LOGICIEL EN ENTREPRISE : Une culture accueillant le changement.*
- Vickoff, J. (2009). *Méthode agile : Les meilleures pratiques compréhension et mise en œuvre.*
- Vygotsky, L. (1980). *Mind in society : The development of higher psychological processes.* Harvard university press.
- Wikipédia. (2019). *Population*. <https://fr.m.wikipedia.org/population>
- Wikipédia. (2020). *Analyse de Contenu*. [https://fr.m.wikipedia.org/wiki/Analyse\\_de\\_contenu](https://fr.m.wikipedia.org/wiki/Analyse_de_contenu)
- Yannick, T. (2007). *Méthodologie d'ingénierie logicielle adaptée à une PME*. Retrieved from <http://diuf.unifr.ch/drupal/sites/diuf.unifr.ch.drupal.softeng/files/teaching/studentprojects/thiessoz/download/rapport.pdf>