



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ ABDELHAMID IBN BADIS - MOSTAGANEM



Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

RAPPORT DE MINI-PROJET
Option : Ingénierie des Systèmes d'Information

Présenté par :

MENAD Hasnia et MEHDI Chima Faiza

THEME :

Développement d'un outil de gestion de progression
de projets basé sur les traits de personnalité

Soutenu le : 19/06/2021

Devant le jury composé de :

SEHABA KARIM MCA Université de Mostaganem Président

LAREDJ MOHAMED ADNANE MCA Université de Mostaganem Examineur

HOCINE NADIA MCB Université de Mostaganem Encadrant

Année universitaire 2020-2021

Dédicaces

*À nos chers parents, pour leur persévérance,
Et pour avoir suscité notre vocation
Et permis d'achever nos études en tant que nous sommes
actuellement,
À nos sœurs, et nos frères
Pour leur soutien moral
Et leur intérêt envers notre travail,
À tous nos amis et collègues,
À tous ceux qui nous ont aidés,
À tous ceux qui nous sont chers,
À tous ceux qui nous avons omis,
Nous dédions cet humble travail.*

*CHIMAA,
HASNIA.*

Remerciements

Au terme de ce travail, nous tenons à remercier Dieu le tout puissant de nous avoir donné le courage, la volonté et la patience pour achever ce travail.

On tient à remercier sincèrement Dr. HOCINE, en tant qu'encadrant, pour ses précieux conseils et son aide, ses orientations et le temps qu'elle nous a accordé pour notre encadrement.

Nous exprimons notre gratitude aux membres de jury qui ont accepté d'examiner notre travail.

Nos remerciements s'étendent également au Chef de département ainsi qu'à tous nos professeurs de la Faculté pour la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience. Enfin, nous adressons nos plus sincères remerciements à tous nos amis, qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Résumé

Les méthodes agiles sont de plus en plus populaires dans le développement logiciel. Les gestionnaires doivent comprendre l'avancement du projet et la qualité des produits sans avoir besoin de documents de développement. Lors des pratiques agiles, les outils de gestion de projets sont fréquemment utilisés afin d'augmenter la vitesse, l'efficacité et la qualité du produit final. L'une des principales tâches est le suivi de la progression de projet (SPM), qui est effectuée pendant le processus de développement du projet. Elle permet de suivre les progrès et de constituer des équipes performantes. Sur le marché aujourd'hui, de nombreux outils sont disponibles pour suivre l'avancement des projets. Cependant, la plupart de ces outils manquent de mécanismes appropriés pour suivre les progrès et former les équipes lors de la gestion de projet. De plus, certains outils ne rapportent pas et / ou n'affichent pas d'informations utiles sur l'état du projet, ni ne fournissent d'informations sur l'état de chaque membre. En conséquence, un outil modulaire de suivi des progrès (PMT) a été développé durant notre projet de Master pour surmonter les limites des outils actuellement utilisés.

Mots clés: Suivi de progression des projets logiciels (SPM), Méthodes Agile

Abstract

Agile methods are increasingly popular in software development. Managers need to understand project progress and product quality without the need for development documents. In agile practices, project management tools are frequently used in order to increase the speed, efficiency and quality of the product. One of the main tasks is Project Progress Tracking (SPM), which is done during the project development process. It makes it possible to monitor progress and build effective teams. On the market today, many tools are available to monitor the progress of projects. However, most of these tools lack mechanisms to track progress and build teams when managing a project. In addition, some tools do not report and / or display useful information on the status of the project, nor do on the status of each member. Consequently, a Modular Progress Tracking Tool (PMT) was developed during our Master project to overcome the limitations of existing tools.

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Processus Scrum	5
Figure 2	Exemple de tableau Kanban (simple)	6
Figure 3	JIRA	15
Figure 4	Asana	16
Figure 5	Trello	16
Figure 6	Wrike	17
Figure 7	Monday	18
Figure 8	BaseCamp	19
Figure 9	Redmine	19
Figure 10	PM-PT	20
Figure 11	Schéma de conception	25
Figure 12	Diagramme de cas d'utilisation	26
Figure 13	Diagramme de classe	27
Figure 14	Diagramme de séquence du système pour l'authentification	28
Figure 15	Diagramme de séquence du système « création de projet »	29
Figure 16	Diagramme de séquence du système « Inscription un membre »	30
Figure 17	Diagramme de séquence du système « Ajouter une tâche »	31
Figure 18	Page de connexion	34

Figure 19	Page d'accueil	35
Figure 20	Ajouter un nouveau projet	35
Figure 21	Afficher la liste des projets	36
Figure 22	Ajouter un nouveau sprint	36
Figure 23	Ajouter une nouvelle tâche	37
Figure 24	Afficher les tâches des projets	37
Figure 25	Données EVA saisies	38
Figure 26	Le tableau de suivi des progrès amélioré	39
Figure 27	Test de personnalité des membres	40
Figure 28	Composition d'équipe	41

Liste des tableaux

Tableau N°	Titre du tableau	Page
Tableau 1	Inventaire De Personnalité HEXACO	12
Tableau 2	Résultat de l'analyse des outils existants	21
Tableau 2	Résultat de l'analyse des outils existants. <i>Suit</i>	22

SOMMAIRE

INTRODUCTION GENERALE.....	1
CHAPITRE I : CONTEXTE ET MOTIVATION	4
1.1 Introduction.....	4
1.2 Méthodes Agile.....	4
1.2.1 Scrum.....	5
1.2.2 Kanban.....	6
1.3 Indicateurs et mesures utilisées pour évaluer la progression et la performance des projets	7
1.3.1 Indicateurs de coûts.....	7
1.3.2 Indicateurs de délais	8
1.3.3 Indicateurs de qualité.....	8
1.3.4 Indicateurs d'efficacité et d'avancement du projet	9
1.4 Composition d'équipe et son influence sur la gestion de projets	9
1.5 Techniques/facteurs de composition d'équipe pour les projets.....	10
1.6 Techniques/facteurs de composition d'équipe pour Agile.....	10
1.7 Conclusion.....	12
CHAPITRE II : ANALYSE DE L'ETAT DE L'ART	13
2.1 Introduction.....	13
2.2 Cadre de recherche et les critères d'analyse	13
2.2.1 Cadre de recherche	13
2.2.2 Critères d'analyse	13
2.3 Analyse des outils de gestion de projet.....	14
2.3.1 JIRA.....	14
2.3.2 Asana	15
2.3.3 Trello.....	15
2.3.4 Wrike	17
2.3.5 Monday	17
2.3.6 BaseCamp	18
2.3.7 Redmine	18

2.3.8 PM-PT (Progress Monitoring Prototype Tool).....	19
2.4 Discussion.....	20
2.5 Introduction à notre contribution de projet de Master	23
2.6 Conclusion	23
CHAPITRE III: CONCEPTION	24
3.1 Introduction.....	24
3.2 Choix de la méthode de conception.....	24
3.3 Démarche de conception.....	24
3.4 Expression et spécification des besoins	25
3.5 Analyse logique.....	26
3.6 Conclusion	31
Chapitre IV: IMPLEMENTATION.....	32
4.1 Introduction.....	32
4.2 Description de l'environnement de développement.....	32
4.2.1 Java entreprise édition	32
4.2.2 Le serveur Apache-Tomcat	33
4.2.3 MySQL (My Structured Query Language)	33
4.2.4 CSS	33
4.2.5 JavaScript.....	33
4.2.6 L'IDE Eclipse	33
4.3 Tableau de bord PMT	34
4.4 Création de projets, sprint et de tâches dans PMT	35
4.5 Suivi de progression de projet dans PMT	38
4.6 Aide à la composition d'équipe dans PMT	40
4.7 Conclusion	41
CONCLUSION GENERALE	42
BIBLIOGRAPHIE.....	43

INTRODUCTION GENERALE

La méthodologie de développement logiciel Agile est parmi les méthodologies les plus utilisées jusqu'à présent, car elle aide les organisations à développer plus rapidement des projets logiciels. Il existe plusieurs méthodes Agile dont les plus populaires sont «Scrum» et «Kanban». Selon une étude récente, la plupart des organisations mentionnent qui utilise cette méthodologie, les projets réussissent 28% plus que les projets gérés avec des méthodes traditionnelles [1]. Il a été rapporté dans d'autres études que l'utilisation de la méthodologie Agile dans le développement de logiciels réduit les coûts et conduit à une meilleure qualité, productivité et satisfaction des entreprises [2]. La principale raison de l'acceptation de cette méthodologie est la capacité d'adapter facilement les objectifs du projet durant sa réalisation et agir sans attendre forcément sa fin.

De nombreux professionnels admettent que l'Agile aide à terminer le projet plus rapidement et que les outils de gestion de projet Agile aident l'équipe à organiser et améliorer leur qualité et accroître l'agilité des projets [3].

La gestion de projet joue un rôle important dans les organisations de développement logiciel car elle nécessite d'avoir suffisamment d'informations et de l'expérience pour prendre des décisions [5]. L'une des tâches essentielles de la gestion de projets logiciels (SPM) est le suivi des progrès qui doit être effectué pendant le processus de développement afin d'avoir un feedback rapide et peut agir en faveur de l'amélioration du résultat du projet. Cela nécessite d'avoir des connaissances, des techniques et des outils permettant de tracer l'information et s'assurer en temps réel que l'avancement du projet se déroule comme prévu [6]. Cependant, cela s'est avéré être un défi majeur pour les organisations de développement de logiciels, où de nombreux projets échouent encore en raison de mesures de temps, du coût et de qualité non respectées [7] [8]. Un pourcentage croissant de développeurs de logiciels utilisent de plus en plus des outils de gestion de projet Agile. Ces derniers fournissent des interfaces facilitent l'allocation des tâches et la planification des objectifs du projet [4].

Problématique

Les outils de gestion de projets Agile existants, sont limités en termes d'aide à la décision. Le chef de projet (ou le Manager) obtient généralement très peu d'informations sur la progression du projet en temps réel, ce qui rend sa tâche difficile. De plus, les outils

existants ne fournissent pas de l'assistance en termes de composition des équipes projet par rapports aux différents attributs des membres qui les composent tels que leurs compétences, préférences et traits de personnalité.

Objectifs du projet

L'objectif de ce projet de Master est de développer un outil d'aide à la gestion et la supervision de la progression des projets de développement logiciel en Agile. L'outil proposé permettra d'aider les Managers dans leur tâche de gestion et offre une meilleure assistance pour changer les objectifs au cours d'un projet et de former les équipes performantes dans le futur. Notre hypothèse de recherche est que la prise en compte de différents attributs des membres lors de la composition d'équipe permet d'améliorer la performance des équipes, ce qui influencera les résultats du projet.

Démarche et méthodologie adoptées

Nous avons suivi une démarche scientifique à travers laquelle nous avons :

1. Étudié le contexte de projet. Tout d'abord, Nous avons étudié le contexte général de la gestion de projet et les méthodes modernes utilisées pour gérer les projets.
2. Fait une analyse approfondie des actions orientées vers la problématique de la gestion du projet. Ceci nous a permis de comprendre les concepts essentiels liés à la gestion des projets ainsi que les différentes métriques utilisées pour évaluer la progression d'un projet. Nous avons également étudié les facteurs utilisés pour la composition des équipes et comment cela peut se traduire dans le cas d'un projet de développement logiciel.
3. Examiné les outils SPM populaires à l'aide de la méthode d'enquête documentaire. Dans cette étape, les outils récupérés ont été étudiés et leurs limites mises en évidence pour identifier les lacunes dans ce domaine qui ont conduit au lancement de cette étude.
4. Suite à l'analyse des travaux, nous avons choisi le langage de modélisation UML pour faire la conception de notre outil de gestion de projet. Il contient l'ensemble des diagrammes nécessaires à la bonne compréhension et cohésion du système, du diagramme de cas d'utilisation au diagramme de classe. Le tout est accompagné des scénarios et des descriptions nécessaires pour développer un outil de gestion de suivi progression de projet logiciel.

5. Enfin, nous avons développé notre outil tout en explorant les différentes caractéristiques et paramètres qui PMT avec tous les outils et l'environnement de développement qui ont été utilisés pour y parvenir.

Organisation du rapport

Ce rapport est organisé comme suit:

- Chapitre I décrit le contexte général de la gestion de projet. Nous définissons les méthodes Agile les plus déployées ainsi que les mesures utilisées pour évaluer la progression et les performances des projets et les facteurs de composition d'équipes.
- Chapitre II présente l'analyse d'état de l'art ainsi que la discussion du résultat d'analyse. Nous introduisons ainsi notre contribution dans ce projet de Master.
- Chapitre III nous présentons Analyse des besoins et conception nous déterminerons les acteurs qui interagissent avec le système à développer, ainsi que les différents modèles de modélisation UML.
- Chapitre IV nous allons présenter les différents outils ainsi que les langages de programmes utilisés comme J2EE (Java 2 Entreprise Edition) Et nous terminons ce chapitre avec l'arborescence de l'application ainsi que quelques interfaces.
- Enfin, nous terminons ce document par une conclusion générale.

CHAPITRE I : CONTEXTE ET MOTIVATION

1.1 Introduction

On nomme « projet », un ensemble d'activités et d'actions organisées en différentes phases dans le but de répondre à un besoin prédéfini et précis, dans un délai fixé et dans la limite d'une enveloppe budgétaire allouée. Un projet mobilise un certain nombre de ressources de natures diverses, matérielles, organisationnelles, logicielles et principalement des ressources humaines [9]. Il existe plusieurs méthodologies d'organisation de projets qui permettraient d'améliorer sa performance et sa qualité.

Les méthodes Agiles sont largement adoptées par les organisations de développement de logiciels. Dans ce premier chapitre, Nous allons présenter le contexte de notre projet. Nous allons décrire notamment les méthodes Agile les plus utilisées. Nous présentons également les différentes mesures utilisées pour évaluer la progression et les performances des projets ainsi que les facteurs de composition d'équipe.

1.2 Méthodes Agile

Agile est une méthodologie de gestion de projet qui décompose les projets en blocs plus petits et gérables appelés itérations. À la fin de chaque itération, qui se déroule généralement sur un intervalle de temps cohérent, un résultat se produit. Contrairement à la gestion de projet Waterfall, l'agilité est strictement séquencée: vous ne commencez pas la conception tant que la recherche n'est pas terminée. Idem, le développement ne commence pas tant que toutes les conceptions ne sont pas approuvées. Avec l'agilité, les développeurs, les concepteurs et les gens d'affaires travaillent simultanément ensemble [10].

La méthodologie Agile repose sur un Manifeste qui représente la spécification du processus de développement qui aide à garantir la qualité de ses résultats. Quatre valeurs et douze principes sont spécifiés dans le Manifeste Agile: «Individus et interactions sur les processus et les outils», «Logiciel de travail sur une documentation complète», «Collaboration client sur la négociation de contrat», «Répondre au changement en suivant un plan » [11]. Les quatre valeurs du Manifeste Agile sont différemment appliquées dans chaque méthodologie pour développer et fournir correctement un logiciel de haute qualité fonctionnelle. Agile est une philosophie de gestion et de projet complète en soi. Mais pour le mettre en œuvre, il est utile en pratique de travailler avec une méthode ou un cadre plus spécifique. Scrum et Kanban sont les méthodes certainement les plus populaires. Scrum, très

CONTEXTE ET MOTIVATION

utilisée dans cette dernière décennie, a été inventé par les mêmes personnes qui ont écrit le Manifeste Agile et est utilisé dans le développement de logiciels et d'autres projets partout dans le monde.

1.2.1 Scrum

Scrum est la méthodologie de gestion de projet de choix de la plupart des équipes de R&D aujourd'hui. Scrum est célèbre - certains pourraient dire infâme - pour des mots à la mode telsque «Sprints», «Scrums», «Backlogs» et «Bumdowns» [12].

L'essence de Scrum est de travailler en petite équipe (pas plus de neuf personnes) et de diviser le travail en phases de deux semaines appelées «sprint» ou «itération». L'équipe se réunit quotidiennement pendant 15 minutes et des «pauses» dirigées par un Scrum Master pour discuter de la situation. Le Scrum Master agit comme un facilitateur dont le travail est d'éliminer les obstacles et d'aider l'équipe à travailler plus efficacement [12].

Scrum ne se concentre pas sur les projets en soi, mais plutôt sur le temps. Il détermine ses objectifs, il attache et il «sprinte» jusqu'à la ligne d'arrivée. Un autre principe clé pour Scrum est que les tests se produisent à plusieurs reprises et en même temps pendant l'itération, ce n'est pas une étape séparée et distincte comme c'est le cas dans une cascade. La figure 1 illustre des itérations ou des sprints dans le processus Scrum [13].

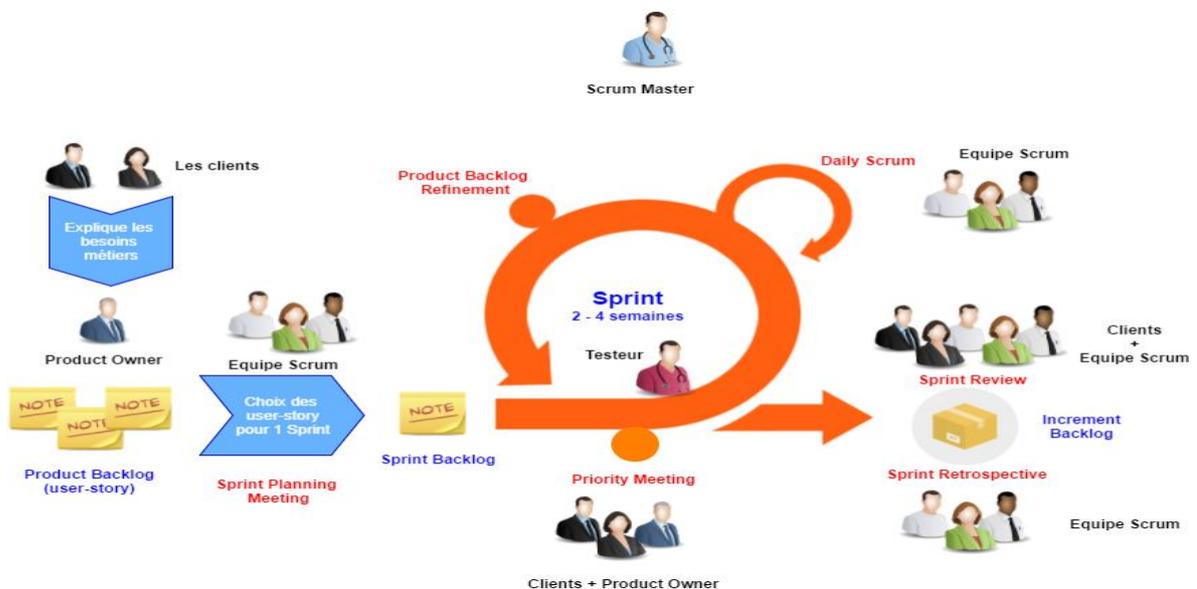


Figure 1 – Processus Scrum [13]

1.2.2 Kanban

En parlant de ramifications Lean, Kanban est l'une d'entre elles. Kanban signifie littéralement «Tableau» en chinois et en japonais, et le tableau Kanban est utilisé pour gérer visuellement les processus.

Un tableau Kanban est divisé en plusieurs colonnes différentes qui représentent les étapes de votre flux de travail. Les étapes peuvent être aussi simples que «À faire», «Travailler dessus» et «Terminé» ou bien plus complexes, selon votre processus. Vous représentez ensuite le travail via des cartes ou des notes autocollantes, en les déplaçant de gauche à droite au fur et à mesure de leur progression dans votre flux de travail. Kanban est essentiellement une chaîne de montage plus conceptuelle, vous permettez d'évaluer visuellement les points d'inefficacité: où se construisent les notes autocollantes? C'est là que vous devez l'améliorer [12].

Comme Agile, Kanban a été spécialement développé pour le développement de logiciels, mais il peut être appliqué à pratiquement tout flux de travail qui suit un processus prévisible [12]. La figure 2 montre un exemple de schéma pour représenter tableau Kanban [14] :

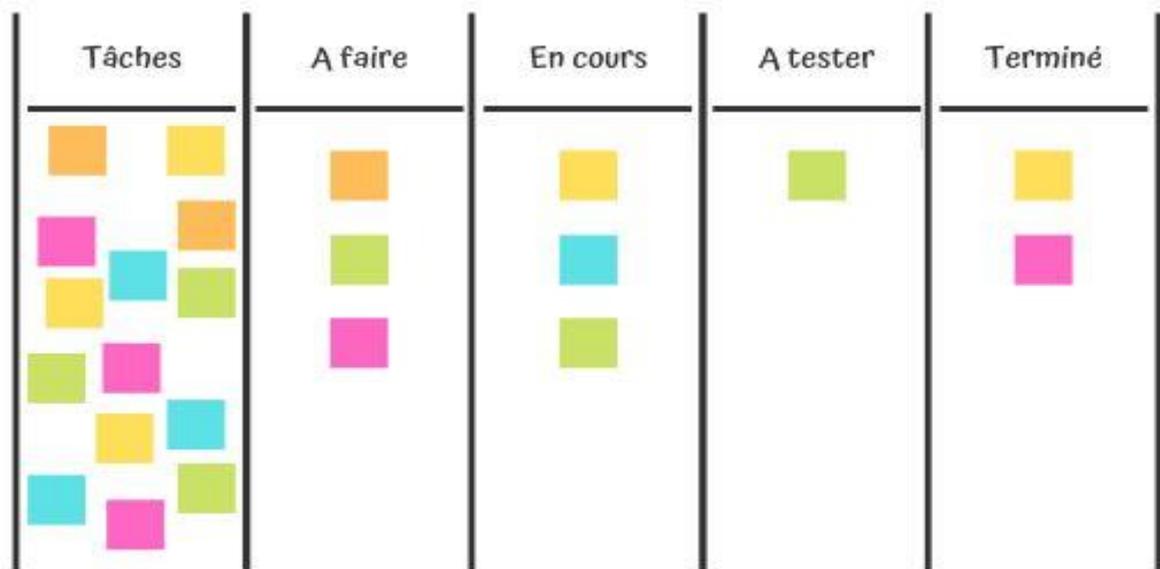


Figure 2 – Exemple de tableau Kanban (simple) [14]

1.3 Indicateurs et mesures utilisées pour évaluer la progression et la performance des projets

Lors de la gestion d'un projet, les délais, le budget, l'étendue du projet et la qualité du travail réalisé doivent toujours être surveillés. Il est essentiel d'utiliser des KPI (Key Performance Indicators), également appelés Key Performance Indicators (KPI). Il est indispensable d'utiliser des KPI (Key Performance Indicators en anglais), aussi appelés indicateurs clés de performance (ICP). Ils sont très utilisés dans le suivi de projet et permettent de mesurer les performances de projets.

Les KPI facilitent la gestion de projet car ils donnent un aperçu rapide de l'état du projet à un instant T. En effet, chaque KPI fournit des informations précises qui permettent d'évaluer la situation à un moment donné et de prendre la décision adaptée.

On distingue quatre catégories de KPI :

- Les indicateurs de coûts : permettent de contrôler que le projet ne dépasse pas le budget alloué.
- Les indicateurs de délais : sont indispensables pour vérifier que le projet respecte les délais et ne subit aucun retard.
- Les indicateurs de qualité : aident à veiller à la qualité du travail fourni.
- Les indicateurs d'efficacité et d'avancement du projet : permettent de savoir si le projet est géré efficacement, si le budget, les ressources et le temps sont utilisés à bon escient, et si la progression du projet est satisfaisante.

1.3.1 Indicateurs de coûts

Différents indicateurs sont utilisés pour estimer le coût, on note en particulier [15] :

- Coût réel : calcul du coût réel du projet. Il se mesure en additionnant toutes les dépenses actuelles du projet.
- Coûts non planifiés : il s'agit de l'ensemble des dépenses réalisées mais non prévues initialement dans le projet.
- Coût des retards: ce KPI permet de connaître les coûts du retard. Il se calcule en faisant la somme des dépenses additionnelles liées au retard comme les heures supplémentaires de main d'œuvre, la maintenance, le dépannage, etc.
- Ecart de coût du projet: la différence entre le coût effectif du projet et le coût initialement prévu. Il est souvent calculé avec la formule suivante :

CONTEXTE ET MOTIVATION

$$Ecat = (\text{coût réel} - \text{coût prévisionnel}) / \text{coût prévisionnel}.$$

- Budget: est intrinsèquement lié aux autres critères de réussite que nous avons déjà mentionnés. D'ailleurs, il constitue généralement le principal critère de réussite d'un projet. L'utilisation de KPI spécifiques permettra de livrer le projet en respectant le budget prévu :
 - Valeur prévue
 - Coût réel
 - Valeur acquise
 - Indice de performance des coûts : Valeur acquise (VA) / coût réel (CR)

1.3.2 Indicateurs de délais

Afin d'estimer les délais, différentes mesures sont utilisées, telles que [15]:

- Taux de retard : ce KPI permet de connaître le pourcentage de retard par rapport au planning initialement prévu.
$$\text{Taux de retard} = (\text{nombre de tâches non réalisées} / \text{nombre de tâches prévues}) \times 100$$
- Durée d'une tâche : utile pour mesurer le temps nécessaire à la réalisation d'une tâche, notamment les tâches récurrentes.
- Ecart de durée : mesure si une tâche ou un jalon est plus long à réaliser par rapport à l'état initial prévu.

$$\text{Ecart de durée} = (\text{durée réelle} - \text{durée initiale}) / \text{durée initiale}$$

1.3.3 Indicateurs de qualité

Il existe plusieurs mesures pour évaluer la qualité d'un projet. On peut noter par exemple [15]:

- La satisfaction du client : un excellent indicateur de qualité qui peut se mesurer grâce à un questionnaire de satisfaction ainsi que par la fidélité du client ou du consommateur.
- Nombre d'erreurs : ce KPI indique le nombre de fois dû refaire une tâche ou retravailler sur un élément. Les erreurs ont un impact sur le budget et le calendrier du projet.
- Les plaintes du client : tout comme la satisfaction du client, il s'agit d'un bon indicateur de la qualité du projet et du travail fourni.

1.3.4 Indicateurs d'efficacité et d'avancement du projet

Parmi les indicateurs les plus importants sont les indicateurs de progression de projets. Ces indicateurs sont de plus en plus utilisés par les chefs de projets afin de les aider à prendre des décisions. Néanmoins, les indicateurs dépendent de type de l'outil utilisé et sa capacité d'analyse et d'extraction d'information en temps réel. Parmi les indicateurs de progression les plus classiques, on trouve [15] :

- Taux d'avancement : comme son nom l'indique, ce KPI permet de savoir quelle est la progression de votre projet.

$$\text{Taux d'avancement} = (\text{tâches accomplies} / \text{tâches prévues}) \times 100$$

- Nombre de tâches ou de jalons réalisés : il s'agit d'un autre bon indicateur de l'avancement du projet.
- Temps passé sur le projet : se fait par la comparaison entre les heures de travail déjà effectuées par rapport aux heures initialement planifiées à tout moment dans le projet pour savoir si l'équipe est efficace. Si la quantité d'heures passées est supérieure à celle prévue, il peut revoir l'estimation du temps nécessaire à la réalisation du projet.

Afin de visualiser la progression des projets, plusieurs outils peuvent être utilisés tels que les graphiques et calendriers Gantt. Il existe des KPI spécifiques pour veiller au respect des délais prévus dans le cadre de projet. On note par exemple :

- Pourcentage de tâches en retard
- Pourcentage des étapes échouées
- Indice de performance du calendrier : Valeur acquise (VA) / valeur planifiée (VP)

1.4 Composition d'équipe et son influence sur la gestion de projets

Le projet étant défini et préparé, le manager de projet va pouvoir constituer une équipe de projet conformément à ses besoins. On appelle « équipe de projet », un petit nombre de personnes œuvrant de manière collective et collaborative, motivé pour atteindre un objectif commun, un résultat visé. Il est important de répartir les rôles et les tâches de chacun en tenant compte de la complémentarité des membres de l'équipe. Notons également que les individus communiquent pour une coopération dans l'action, ils adoptent un fonctionnement organisé par des règles et une autorité qui coordonne les efforts [9].

Généralement, les tâches d'un projet nécessitent des équipes multi-compétences afin de réaliser l'intégralité des activités du projet. Ces groupes multi-compétences réunissent des

CONTEXTE ET MOTIVATION

individus en provenance de départements fonctionnels, chaque département étant organisé selon des compétences spécifiques, relativement homogènes. Le rôle du chef de projet est de définir les relations entre les tâches, les compétences requises et parallèlement de gérer le budget, la qualité du produit et les délais. De ce fait la constitution d'une équipe de projet devient un " challenge" pour le chef de projet [16].

1.5 Techniques/facteurs de composition d'équipe pour les projets

Afin de mener à bien un projet, il est important de définir les ressources humaines nécessaires qui permettront tout au long du cycle de vie du projet de prendre les bonnes décisions. Pour atteindre les objectifs demandés, bâtir une équipe efficace passe par :

- Réunir les compétences utiles.
- Définir les rôles et responsabilités de chacun et les diffuser à tous les intervenants.
- Créer un véritable esprit de groupe en pratiquant la création collective avec neutralité de propos et tolérance d'idées.
- Canaliser l'énergie de chacun, chaque intervenant doit connaître exactement les objectifs et les limites de son champ d'action.
- Faire corriger systématiquement et collectivement les erreurs et dysfonctionnements.

1.6 Techniques/facteurs de composition d'équipe pour Agile

La formation de l'équipe est la première étape de la gestion de projets [17]. C'est un processus qui commence par identifier et sélectionner les individus avec un certain ensemble de caractéristiques requises, puis les rassembler pour former une équipe en fonction de son contexte d'utilisation [18]. Les caractéristiques et comportements des membres doivent être considérés pour former une équipe efficace. Il existe deux approches dans la formation d'équipe [19] :

- **L'approche descendante** : est la méthode conventionnelle de formation d'équipe dans laquelle les membres de l'équipe sont sélectionnés, formés et contrôlés par la direction supérieure (chef de projet).
- **L'approche ascendante** : se fait entre les membres de l'équipe eux-mêmes, tout en préservant les besoins contextuels tels que les connaissances, les compétences et les capacités requises par chaque projet de développement différent.

CONTEXTE ET MOTIVATION

Généralement, la composition de l'équipe peut être homogène ou hétérogène. Homogène signifie que les membres de l'équipe ont les mêmes caractéristiques tandis que hétérogène signifie qu'il existe une différence significative de caractéristiques entre les membres de l'équipe.

Puisque le Manifeste Agile stipule qu'une équipe Agile doit être transverse (un mélange de compétences dans une équipe), la composition de l'équipe doit être hétérogène. De plus, un certain degré d'hétérogénéité peut garantir la performance de l'équipe et maintenir l'équipe en synergie. Dans une composition hétérogène, les membres de l'équipe ont des antécédents différents mais ils doivent être équilibrés en termes d'expériences, de compétences techniques, de connaissances du domaine, de taille de l'équipe, de diversité (sexe, race, culture, modes de pensée et comment ils résolvent les problèmes et prennent des décisions [20]).

L'une des premières étapes de la formation d'une équipe est la sélection des membres de l'équipe ayant les caractéristiques appropriées nécessaires à une composition précise. Dans la GRH pour un projet spécifique, il existe plusieurs critères de sélection lors de la constitution d'une équipe projet. Comme indiqué dans le manuel de gestion de projet du livre des connaissances (PMBOK), les critères pris en considération lors de la sélection des membres de l'équipe de projet sont très similaires aux caractéristiques de composition mentionnées précédemment telles que les connaissances, l'expérience, la capacité, la compétence, l'attitude, la disponibilité, coût et facteurs internationaux [21].

- La disponibilité fait référence à la capacité d'un membre de l'équipe à s'engager dans l'équipe de projet jusqu'à ce qu'elle se termine.
- Les critères de coût sont, de vérifier si le coût d'acquisition du membre de l'équipe est toujours dans le budget alloué.
- Les facteurs internationaux font référence à l'emplacement, au fuseau horaire et aux capacités de communication du membre.

Des études récentes sur la composition des équipes des développeurs de logiciels se sont centrées sur les caractéristiques de la personnalité des membres [22]. La personnalité signifie la façon de penser, de ressentir et de se comporter. Les traits de personnalité sont différents d'un individu à un autre. Chaque personne réagit différemment aux situations et rôles attribués. En identifiant une personnalité appropriée pour un rôle spécifique, une combinaison équilibrée de membres peut être déterminée pour une équipe [23]. Par exemple,

CONTEXTE ET MOTIVATION

une équipe Scrum comprend des rôles Scrum Master (SM), Product Owner (PO) et l'équipe de développement. Ainsi, une personnalité appropriée doit être identifiée pour chacun des rôles. Par exemple pour le rôle SM, la personne sélectionnée pour ce rôle doit avoir de la patience. Pour les rôles PO et SM, la personne doit être organisée.

Les traits de personnalité peuvent être évalués à l'aide des questionnaires et des outils d'évaluation comme par exemple HEXACO [24]. Ce dernier comprend six traits, à savoir l'honnêteté-humilité, l'émotivité, l'extraversion, l'agréabilité, la conscience et l'ouverture à l'expérience. Les caractéristiques de chaque trait de personnalité sont données dans **tableau 1**.

Tableau 1 – Inventaire De Personnalité HEXACO

Traits de personnalité	Traits étroits
Honnêteté – Humilité	La sincérité- Justice- Évitement de la cupidité- Modestie
Émotivité	Peur- Anxiété- Dépendance- Sentimentalité
Extraversion	Estime de soi sociale- Audace sociale- Sociabilité- Vivacité
Agréabilité	Le pardon- Douceur- Souplesse- Patience
Conscience	Organisation – Diligence – Perfectionnisme- Prudence
Ouverture	Appréciation esthétique – Curiosité - La créativité - Non-conventionalité
Échelle interstitielle	Altruisme

1.7 Conclusion

Nous avons présenté dans ce chapitre le contexte de notre projet. Nous avons décrit la méthodologie Agile ainsi que les différents indicateurs et mesures utilisées pour aider à l'évaluation de la progression d'un projet ainsi que la composition des équipes. Dans le prochain chapitre, nous présentons le résultat de l'analyse de l'état de l'art des outils de gestion de projets pour l'aide au suivi et à la composition d'équipes-projets.

CHAPITRE II : ANALYSE DE L'ETAT DE L'ART

2.1 Introduction

Les systèmes de gestion de projet actuellement utilisés fonctionnent en s'appuyant sur les données collectées par l'utilisateur. Les projets sont planifiés et gérés en fonction des dates de début et de fin fournies par l'utilisateur selon la décision du chef de projet. Afin d'éviter l'utilisation de données basées sur l'expérience, un système de gestion de projet facilitant la collecte et le calcul des données peut être développé [26]. En considérant le système dans son ensemble, il est possible de regrouper les paramètres qui seraient utilisés pour sa gestion en trois groupes: paramètres basés sur le projet, paramètres basés sur les employés et les paramètres basés sur les tâches.

Différents outils d'aide à la gestion de projets de développement logiciel en Agile sont proposés. Ces outils proposent différentes fonctionnalités et services qui aident à la fois le Manager et les membres. Nous présentons dans ce chapitre le résultat de notre analyse de l'état de l'art sur les outils de gestion de progression et de projets de développement logiciel. Nous présentons d'abord les critères de recherche et d'analyse puis nous décrivons les outils analysés. Nous clôturons ce chapitre par une discussion et une introduction à notre contribution dans ce projet de Master.

2.2 Cadre de recherche et les critères d'analyse

2.2.1 Cadre de recherche

Le cadre de recherche était les outils de gestion de projet utilisés pour suivre et évaluer la progression et performance d'un projet Agile. Nous avons utilisé Google Scholar et Google comme moteurs de recherche. Les mots clés utilisés sont : outils de gestion de projet Agile, composition d'équipes Agile, progression du projet, performance du projet.

2.2.2 Critères d'analyse

Nous nous sommes basé sur un ensemble de critères pour comparer les différents outils existants pour répondre à notre problématique de recherche. Les principaux critères de notre analyse étaient :

- **Existence d'un moyen de suivi de progression** : Pour chaque outil sélectionné, nous allons déterminer si cet outil possède des fonctions permettant le suivi de progression de projet. Si c'est le cas nous allons analyser l'ensemble de métriques utilisées pour cette

finalité. En particulier nous avons analysé les métriques liées aux :

- **Les délais** : Suivi du temps à chaque étape du projet pour terminer le projet à temps.
- **Le budget** : La question se pose : avez-vous dépassé le budget alloué au projet ?
- **Existence d'un moyen d'aide à la composition d'équipe** : Nous allons décrire pour chaque outil analysé si cet outil permet d'aider le chef de projet (ou le Manager) à constituer des équipes. On s'intéresse en particulier à savoir comment cet aide est effectué et sur la base de quels paramètres ou indicateurs ?
- **Indicateurs/mesures donnés au manager pour obtenir des informations sur l'état de chaque membre** : Pour chaque outil analysé, nous allons décrire les informations et les indicateurs que l'outil donne au Manager sur les membres et comment ses informations sont utilisées ?

2.3 Analyse des outils de gestion de projet

Dans cette recherche, nous étudierons les outils de gestion de projet Agile populaires utilisés par les professionnels. De plus, les limites de ces outils sont déterminées par les critères d'analyse.

2.3.1 JIRA

Jira est un système de suivi des projets professionnel largement utilisé de la société Atlassian. Il s'agit d'un outil de planification et de collaboration basé sur le Web utilisé pour planifier les flux de travail et suivre les progrès (voir Figure 3).

Jira Agile est une version de l'outil Jira anciennement connu sous le nom de Green Hopper. C'est un plugin qui ajoute la gestion de projet Agile à n'importe quel projet Jira. De plus, il dispose d'un tableau de tâches numérique qui peut fournir divers flux de travail personnalisables pour la gestion des tâches et stimuler l'amélioration progressive [27]. Cependant, Jira est un outil assez complexe qui nécessite une courbe d'apprentissage en termes de suivi et de paramétrage des workflows. Par conséquent, son utilisation nécessite un effort administratif important [28].

De plus, il n'a pas de fonctionnalité de recherche avancée car il ne prend en charge que les correspondances exactes [39] et présente un défi important dans le suivi de la progression car il ne prend pas en charge une fonction de planification comme indiqué par [30], Il n'aide

pas le manager de projet dans la sélection de l'équipe de gestion de projet optimale en termes de caractéristiques personnelles.

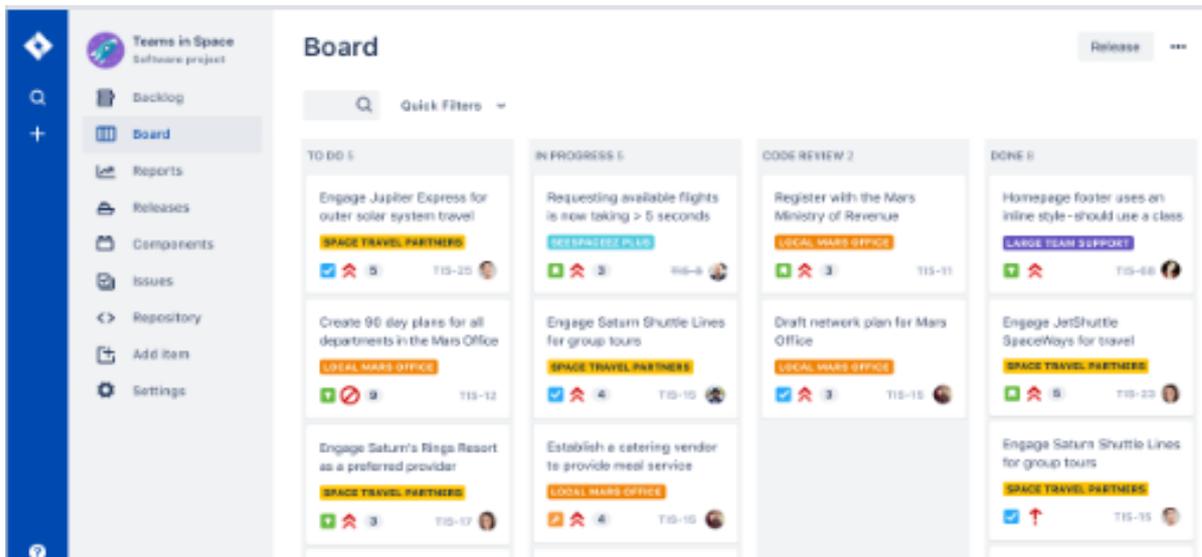


Figure 3 – JIRA

2.3.2 Asana

Asana est l'un des meilleurs outils de gestion des tâches, permettant aux équipes de contribuer, de coordonner et de suivre le développement des missions. Chaque devoir peut inclure une note, des remarques, des notes et un marqueur [31]. Asana propose une interface intuitive et un service très complet, basé sur l'efficacité et la souplesse d'utilisation : planifier chaque étape, attribuer les tâches, créer un planning, partager des fichiers et garder un œil sur les dates clés de projet grâce à un calendrier (voir Figure 4).

Asana aide les équipes à organiser leur travail, des tâches quotidiennes aux initiatives stratégiques, mais ne contient pas de moyen de choisir la bonne équipe pour assurer l'organisation et la collaboration entre l'équipe.

2.3.3 Trello

Trello, un outil basé sur l'approche Kanban, est l'un des outils de gestion de projet les plus populaires. Il est pris en charge par les plates-formes mobiles. Les projets sont définis par des tableaux comprenant des listes, où chaque liste a des cartes progressives, utilisées comme glisser-déposer [31]. D'autres fonctionnalités intéressantes peuvent être répertoriées, telles que l'intégration avec d'autres applications et l'utilisation de listes de contrôle, la possibilité d'écrire des commentaires, des notes et d'ajouter des pièces jointes.

ANALYSE DE L'ETAT DE L'ART

Trello simplifie la collaboration en organisant et en effectuant le suivi de toutes les tâches, fichiers et informations d'équipes en un seul endroit (voir Figure 5). Cependant, l'outil Trello n'a pas la capacité de suivre les projets et cet outil ne permet pas au gestionnaire de suivre l'avancement global du projet, Trello ne contient pas de moyen de sélectionner l'équipe en termes de capacités pour s'assurer que les objectifs du projet sont atteints.

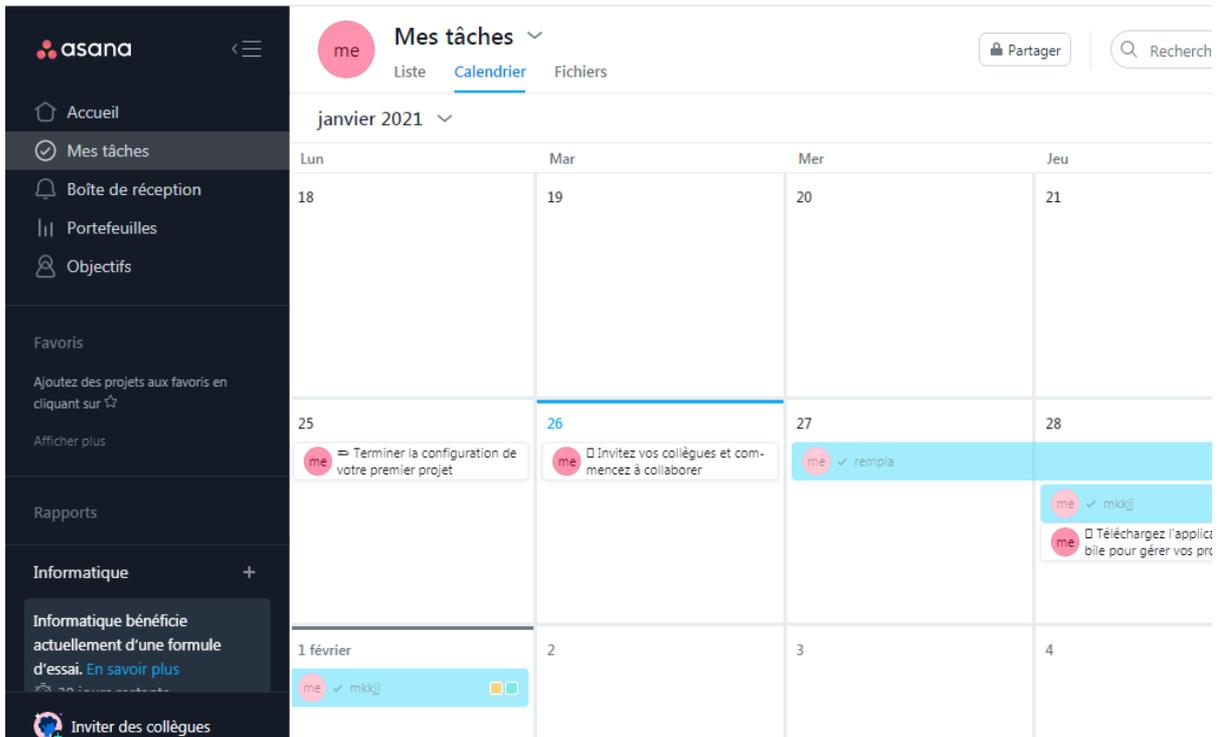


Figure 4 – Asana

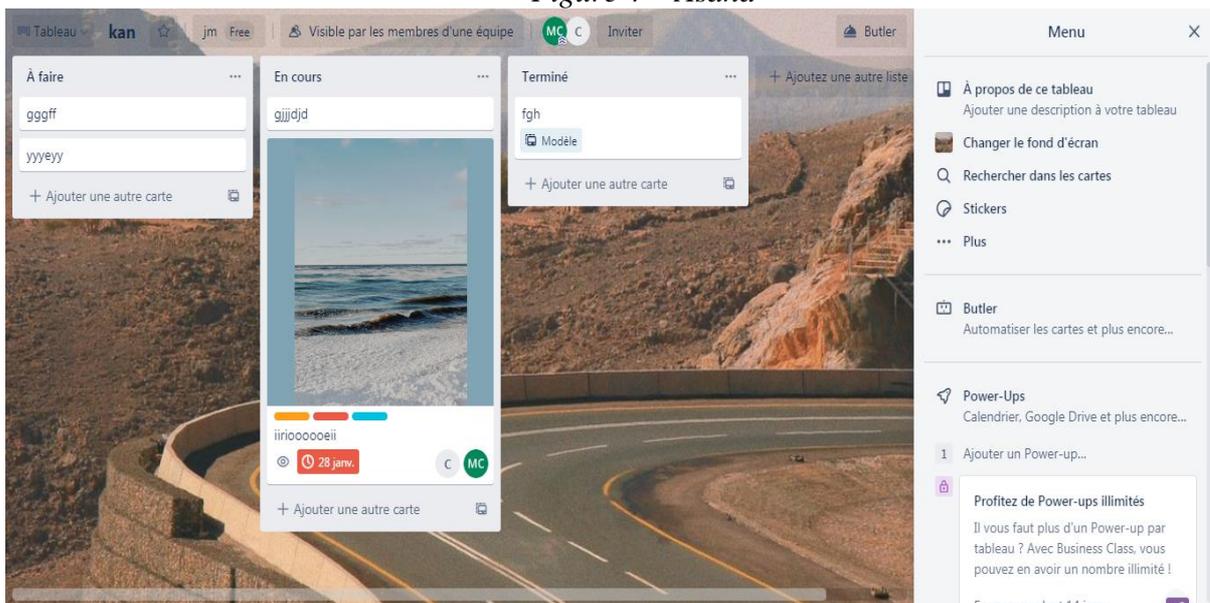


Figure 5 – Trello

2.3.4 Wrike

Semblable aux outils précédents, Wrike permet également la personnalisation et la collaboration entre les membres de l'équipe en offrant la flexibilité nécessaire pour administrer plusieurs projets et équipes. Au cours du processus Agile, les développeurs recevront des informations récentes et exactes. La figure 6 donne un aperçu sur l'interface utilisateur [31].

L'outil fournit également une intégration de messagerie via la gestion de projet [31]. Wrike permet de planifier, hiérarchiser, discuter et garder trace du travail et des progrès en temps réel. Néanmoins, l'outil Wrike n'a pas la capacité de choisir la bonne équipe pour un projet.

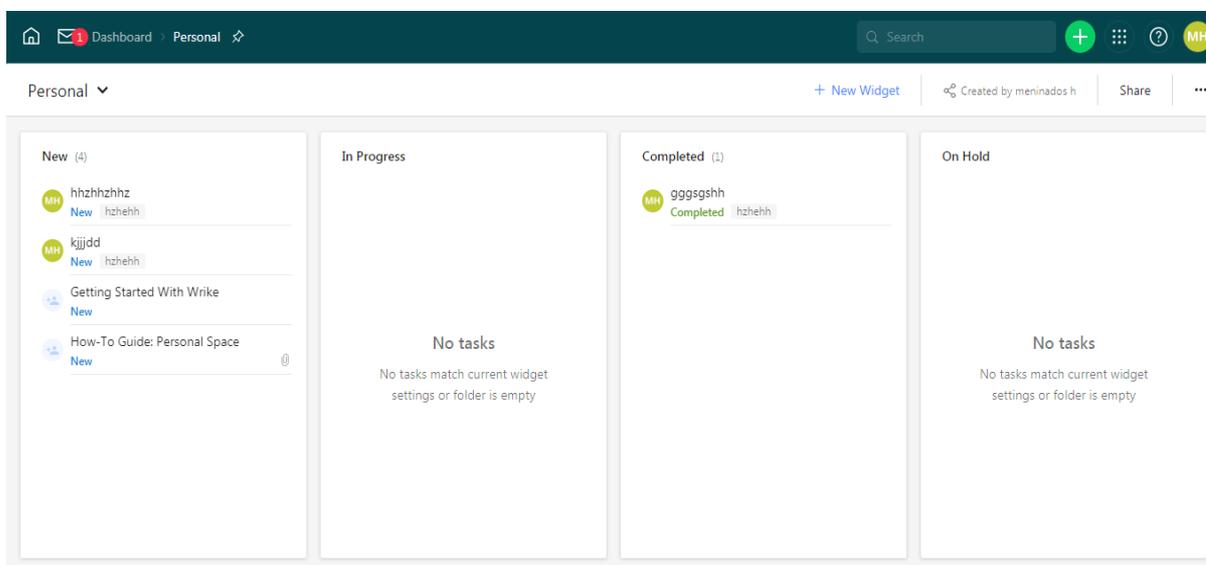


Figure 6 – Wrike

2.3.5 Monday

Monday.com permet aux équipes à distance de planifier, organiser et suivre au quotidien leur travail sur une seule et même plateforme. Considérée comme une référence sur le marché des outils de gestion de projet, elle est déjà adoptée par des équipes de grandes entreprises telles que Carrefour, eBay, Gameloft ou encore Deezer. Figure 7 illustre un exemple d'interface utilisateur pour la gestion des tâches.

La plateforme monday.com facilite le télétravail en centralisant la gestion des tâches de chaque collaborateur sur une seule et même interface. Les courses de travail (jours, semaines, mois) peuvent être définies, assigner et commenter des tâches mais aussi définir un statut et une priorité pour chacune de ces tâches. Un aperçu calendrier donne une vision plus

globale du projet et des tâches de l'ensemble de l'équipe sur les semaines à venir [32]. Selon le forfait qu'il choisit, l'administrateur du compte peut accéder à la gestion des droits, au suivi du temps (sans service de facturation) et à la création d'un tableau de bord permettant de visualiser l'évolution du projet et du budget alloué.

Cet outil permet de créer les flux de travail ou de choisir parmi de nombreux espaces de travail pour synchroniser, planifier, organiser et suivre les projets d'équipe. De plus, il ne contient pas de moyen de choisir la bonne équipe pour assurer l'organisation et la coopération entre l'équipe.

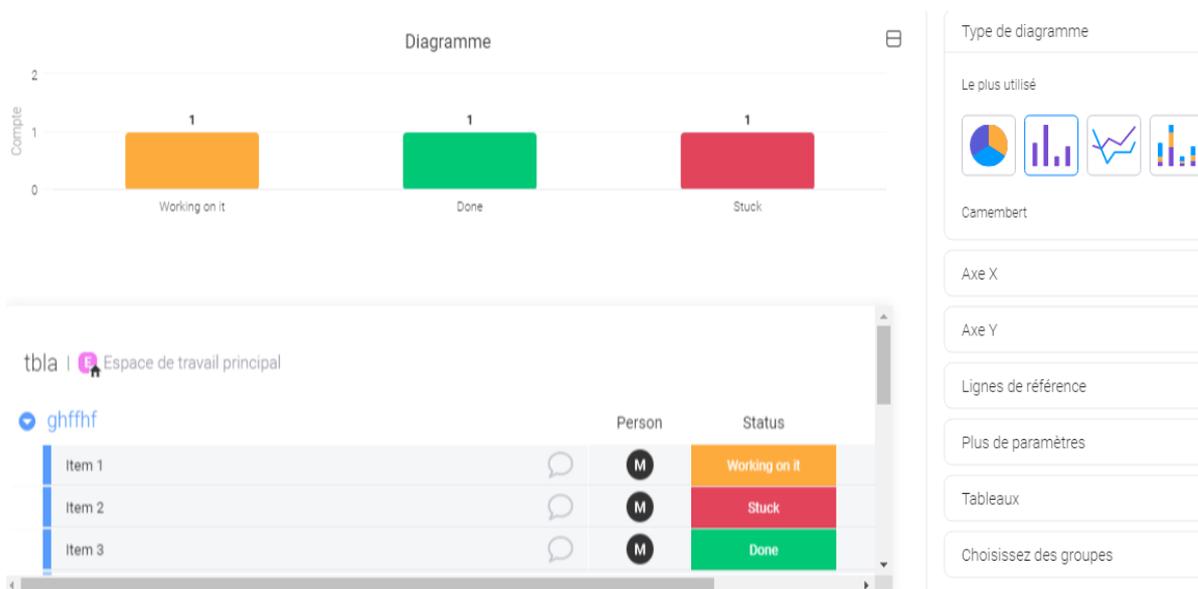


Figure 7 – Monday

2.3.6 BaseCamp

BaseCamp est un outil utile utilisé pour les collaborations et la gestion de projets ainsi que pour la communication à l'échelle de l'entreprise. C'est un outil modifiable avec une gestion des tâches très simple [33]. En raison de son extrême simplicité, la préparation de rapports plus avancés doit être effectuée manuellement par les membres de l'équipe et les gestionnaires, ce qui augmente la probabilité d'erreurs humaines [34]. De plus, BaseCamp n'autorise pas l'attribution de délais spécifiques pour les tâches ou projets [35], Il n'aide pas le chef de projet à sélectionner l'équipe optimale pour gérer le projet.

2.3.7 Redmine

Pavić [36] a décrit Redmine comme un outil flexible qui prend en charge la gestion de la configuration logicielle ainsi que plusieurs langues et bases de données multiples. Figure 9

montre un exemple de l'interface utilisateur dans Redmine. Bien que Redmine aide à gérer l'état actuel du projet et à représenter graphiquement les changements, il ne fournit aucun support particulier pour les méthodes Agile [37].

Redmine n'a pas de fonction de rapport et ne prend pas en charge la génération automatique de graphiques à des fins d'analyse des progrès [38], Il ne contient pas de moyen de sélectionner l'équipe en termes de compétences pour s'assurer que les objectifs du projet sont atteints.

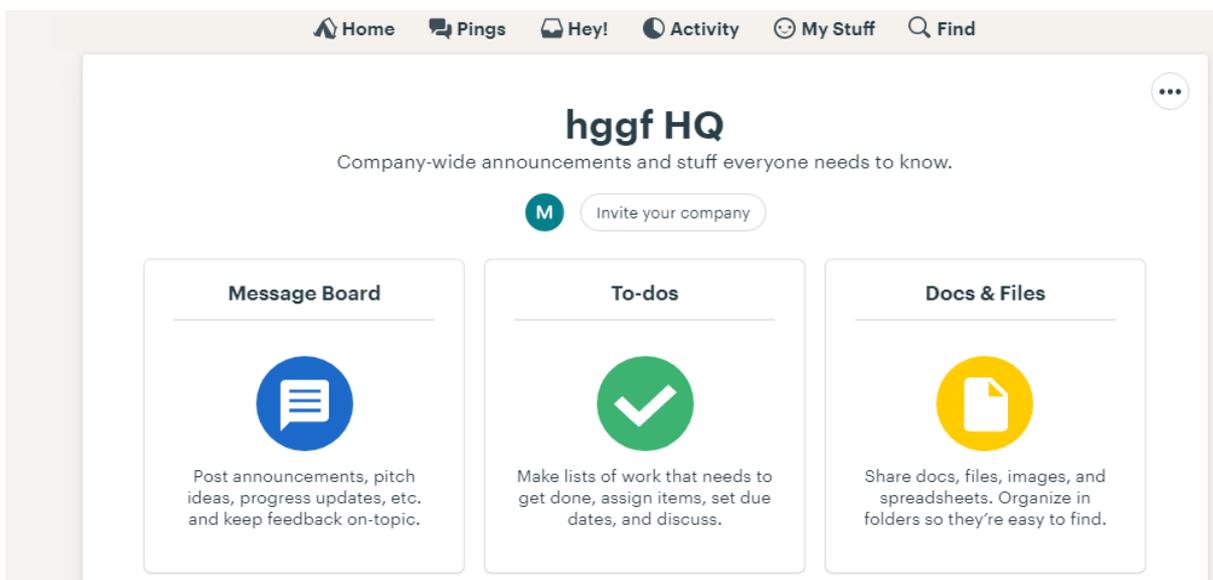


Figure 8 – BaseCamp

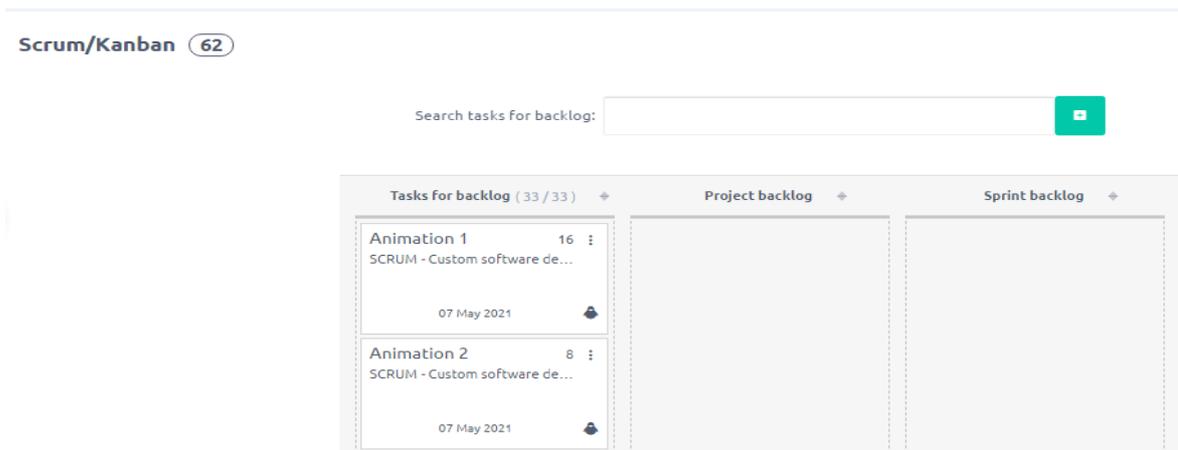


Figure 9 – Redmine

2.3.8 PM-PT (Progress Monitoring Prototype Tool)

PM-PT est un prototype d'outil de suivi des progrès qui, permet aux chefs de projets et aux développeurs de logiciels de suivre facilement l'avancement de leurs projets logiciels.

PM-PT a été conçu sur la base du modèle de tâche de surveillance de projet logiciel amélioré de la méthode Kanban Agile (i-KAM). En particulier, PT PM peut suivre l'avancement des projets grâce à un mécanisme alternatif qui utilise la méthode d'analyse de la valeur des gains (EVA). De plus, il peut générer les limites WIP (Work In Progress) optimales pour chaque étape du tableau Kanban. La figure 10 fournit un aperçu sur l'outil PM-PT [39].

PM-PT améliore la visualisation du flux de travail en fournissant des informations utiles qui aident les chefs de projet à prendre des décisions significatives concernant l'avancement du projet [39]. Cependant, il n'aide pas le chef de projet à identifier la bonne équipe pour gérer le projet.

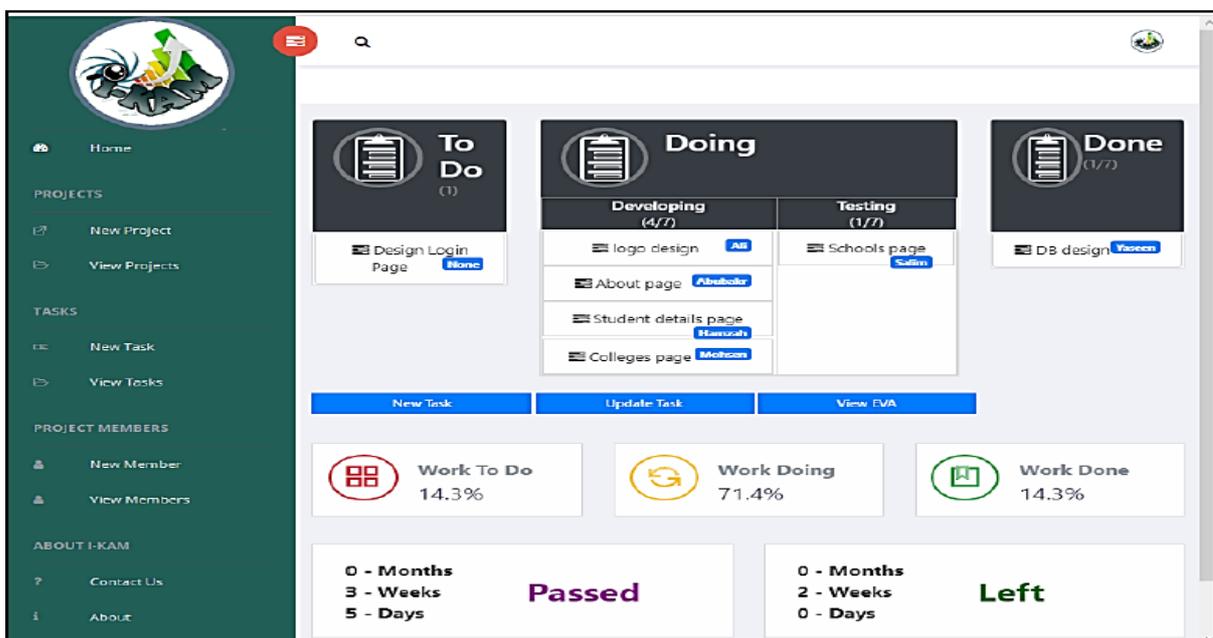


Figure 10 – PM-PT

2.4 Discussion

Comme décrit dans les sections précédentes, de nombreux outils de gestion de projet sont présents. Certains d'entre eux sont spécifiques, utilisés pour des types de méthodologies Agile telles que Scrum, Kanban, XP, tandis que d'autres sont plus génériques, utilisés pour tout type de méthodes Agile ou tout secteur. Nous avons résumé tous les outils dans un tableau par critères d'analyse.

Tableau 2 – Résultat de l'analyse des outils existants

Outils	Moyen de suivi de progression	Moyen d'aide à la composition d'équipe	Informations sur l'état de chaque membre
JIRA	<ul style="list-style-type: none"> - A un défi important dans le suivi des progrès - Ne prend pas en charge la fonction de planification 	<ul style="list-style-type: none"> - Il n'aide pas le manager de projet dans la sélection de l'équipe de gestion de projet optimale 	<ul style="list-style-type: none"> - Suivre les performances de chaque membre en exploitant en temps réel les données visuelles des rapports. - Un workflow dans Jira Software est représenté sur un tableau numérique et affiche les états et transitions de chaque tâche.
Trello	<ul style="list-style-type: none"> - Pas ou peu de gestion de ressources, de temps ou d'équipe. 	<ul style="list-style-type: none"> - Ne contient pas de moyen de sélectionner l'équipe en termes de capacités 	<ul style="list-style-type: none"> - L'utilisation est limitée aux projets ponctuels et non à la productivité quotidienne (gestion de tâches en équipe)
Asana	<ul style="list-style-type: none"> - Ils ne permettent pas de gérer des projets complexes avec toutes leurs dimensions (budgets, etc.). 	<ul style="list-style-type: none"> -Ne contient pas de moyen de choisir la bonne équipe 	<ul style="list-style-type: none"> - Fonction de rapport
Monday	<ul style="list-style-type: none"> - Besoin d'améliorer le mécanisme de suivi des progrès 	<ul style="list-style-type: none"> -Ne contient pas de moyen de choisir la bonne équipe 	<ul style="list-style-type: none"> - N'aide pas le manager à visualiser les informations sur l'état de chaque membre
Wrike	<ul style="list-style-type: none"> - Ne dispose pas d'un mécanisme de suivi qui a un impact négatif sur l'avancement du projet. 	<ul style="list-style-type: none"> -N'a pas la capacité de choisir la bonne équipe pour un projet 	<ul style="list-style-type: none"> - la préparation de rapports doit être effectuée manuellement par les membres de l'équipe et les gestionnaires, ce qui augmente la probabilité d'erreurs humaines

ANALYSE DE L'ETAT DE L'ART

Redmine	<ul style="list-style-type: none"> - Ne fournit aucun support particulier pour les méthodes Agile - Ne prend pas en charge la génération automatique de graphiques à des fins d'analyse des progrès 	<ul style="list-style-type: none"> -Ne contient pas de moyen de sélectionner l'équipe en termes de compétences 	<ul style="list-style-type: none"> - N'a pas de fonction de rapport
BaseCamp	<ul style="list-style-type: none"> - Nécessite la création manuelle de rapports avancés. Ainsi, des erreurs d'êtres humains pourraient se produire - Manque de fonctionnalités de rapport - Ne permet pas d'attribuer des délais spécifiques pour les tâches ou projets 	<ul style="list-style-type: none"> - N'aide pas le chef de projet à sélectionner l'équipe optimale pour gérer le projet 	<ul style="list-style-type: none"> - Fonction de rapport
PM-PT	<ul style="list-style-type: none"> - Facilite la gestion de projet en termes de suivi de l'avancement du projet, d'estimation réelle du projet et d'analyse en temps réel de l'ensemble du projet. Cependant, il ne dispose pas de mécanisme d'analyse des risques. 	<ul style="list-style-type: none"> - N'a pas la capacité de choisir la bonne équipe pour un projet 	<ul style="list-style-type: none"> - N'a pas la capacité de suivre l'état de chaque membre

Premièrement, les outils SPM populaires ont été examinés à l'aide de la méthode d'enquête documentaire. Dans cette phase, les outils récupérés ont été étudiés et leurs limites ont été mises en évidence pour identifier les lacunes dans ce domaine qui ont conduit au lancement de cette étude.

Comme le montre le tableau II, La plupart de ces outils manquent de mécanisme de suivi des progrès et la visualisation inadéquate des informations cruciales et des rapports utiles pour l'avancement du projet. Elles manquent aussi de mécanisme de sélection de l'équipe de projet en fonction des capacités cognitives et personnelles qui aident à former une équipe hautement qualifiée pour exécuter le projet et atteindre l'objectif prévu.

La gestion de projet logiciel (SPM) nécessite des informations précises et de l'expérience pour prendre une décision. L'une des tâches difficiles de la SPM est de savoir comment créer des équipes efficaces avec une attribution appropriée des tâches et ensuite suivre l'avancement du projet afin de garantir le respect des objectifs et des délais prévus.

2.5 Introduction à notre contribution de projet de Master

L'objectif de ce projet de Master est de développer un outil SPM qui aide le manager à construire un travail d'équipe efficace. Non seulement les expériences et les compétences des membres de l'équipe à prendre en compte, mais aussi leurs traits de personnalité. L'outil aide également le gestionnaire à obtenir et à suivre l'avancement du projet grâce à un mécanisme de suivi avec la possibilité de visualiser des informations cruciales. L'outil SPM est développé en J2E. Les fonctionnalités clés de notre outil sont:

- Cet outil permet au Manager et aux membres de créer des projets et signaler leur progression.
- Cet outil permet au Manager de suivre la progression individuelle de chaque membre : avoir une traçabilité des tâches effectuées dans les projets, des informations sur sa personnalité, sa progression individuelle afin de mieux guider le manager pour la composition de futurs équipes.
- Gestion de tâches : créer les tâches, affecter ces tâches aux membres de projet et de travailler sur les points d'action du projet.
- Suivi de projet ; suivi de l'avancement de projet en temps réel et suivi du budget sur des tableaux de bord et des diagrammes.

2.6 Conclusion

Nous avons présenté dans ce chapitre le résultat de l'analyse de l'état de l'art sur les outils de gestion de projets Agile dans le cadre de développement logiciel. Cela nous a permis d'introduire notre contribution par rapport à la problématique de ce projet de Master.

CHAPITRE III: CONCEPTION

3.1 Introduction

Dans ce chapitre nous allons présenter la partie conceptuelle de notre outil de gestion de progression de projets. La méthode RUP à été adoptée ainsi que le langage de modélisation UML.

3.2 Choix de la méthode de conception

Pour la réalisation de ce projet nous avons utilisé UML avec les diagrammes travaillés avec Visual paradigm online (<https://online.visual-paradigm.com>).

UML est un langage de modélisation unifié « Unified Modeling Language» à base de diagrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la «conception orienté objet». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine de l'informatique [43].

UML est l'accomplissement de la fusion de précédents langages de modélisation objet: Booch, OMT, OOSE.UML est à présent un standard défini par l'Object Management Group (OMG).

- Les diagrammes de structure (structure diagrams) ou diagrammes statiques (static diagrams) rassemblent par exemple des diagrammes de classe (class diagram) qui est la représentation des classes dont une modélisation orienté objet.
- Diagrammes comportementaux ou diagrammes dynamiques : Les diagrammes d'interaction (interaction diagrams) ou diagrammes dynamiques (dynamic diagrams) rassemblent par exemple les diagrammes de séquence (sequence diagram) qui est la représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

3.3 Démarche de conception

Il est évident que le succès de développement logiciel est fonction de la bonne utilisation de la méthode, mais se succès dépend aussi et surtout de la façon d'utilisation de cette méthode à l'intérieur du cycle de développement logiciel. On ne peut pas réaliser l'application de notre projet d'une manière simple et facile sans l'utilisation d'une méthode d'analyse convenable. En effet, tout travail informatique doit être réalisé avec une méthodologie qui aide les réalisateurs à mieux conduire leurs projets. La méthode utilisée

pour la conception de notre projet c'est la méthode PU [44].

Le processus unifié (PU) ou UP (anglais : unified process), est une méthode de développement pour les logiciels orientés objets. C'est une méthode générique, itérative et incrémentale, contrairement à la méthode séquentielle Merise. PU vient compléter la systémique des modèles UML. Elle est le résultat final d'une évolution de l'approche d'Ericsson qui est au fondement d'une des premières méthodes de développement pour applications orientées objets [45]. Selon les principes du processus RUP et les besoins de notre système d'information du projet nous avons utilisé une démarche de modalisation définie par une séquence d'étapes montrées dans la figure [11]:

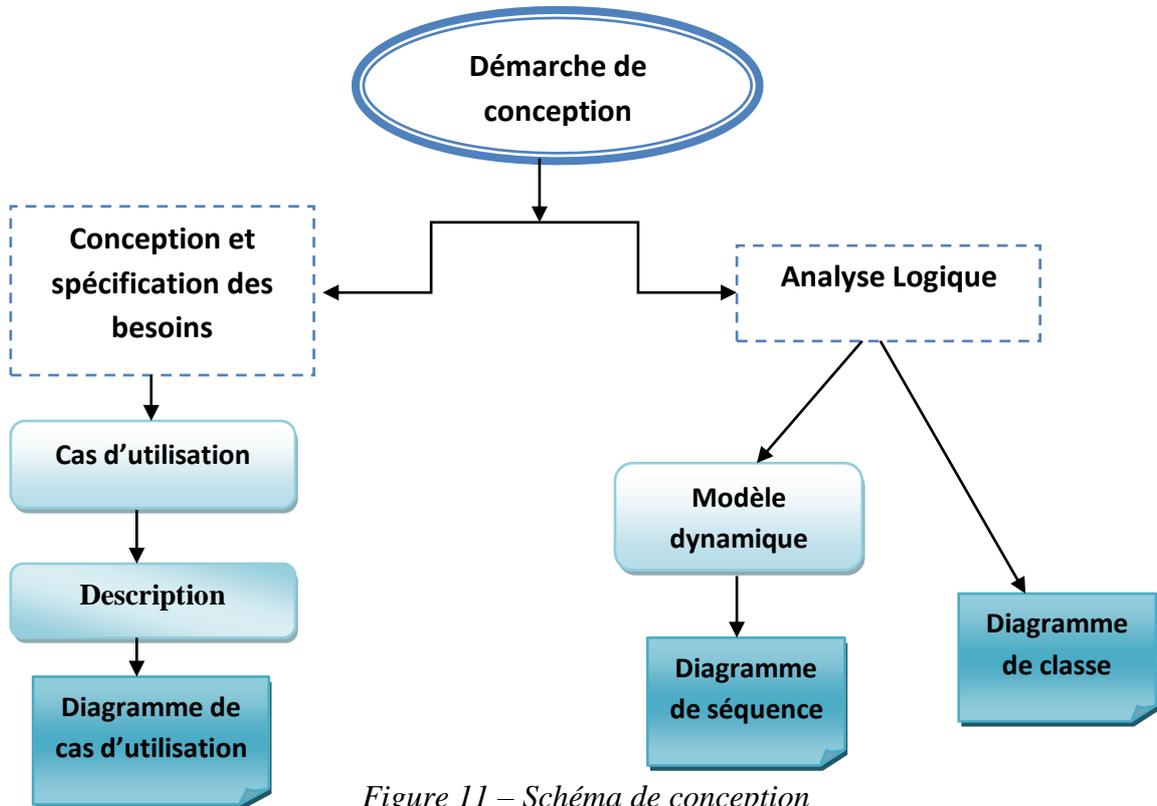


Figure 11 – Schéma de conception

3.4 Expression et spécification des besoins

Ce cycle consiste à réaliser la délimitation des besoins fonctionnels, il agit aussi pour préparer le terrain à l'analyse orientée objet. Nous présenterons le diagramme de cas d'utilisation général pour notre outil proposé. Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel [46]. Nous allons désormais parler de l'interaction entre les acteurs et notre système : il s'agit de décrire la chronologie des actions qui devront être réalisées par les

acteurs et par le système lui-même.

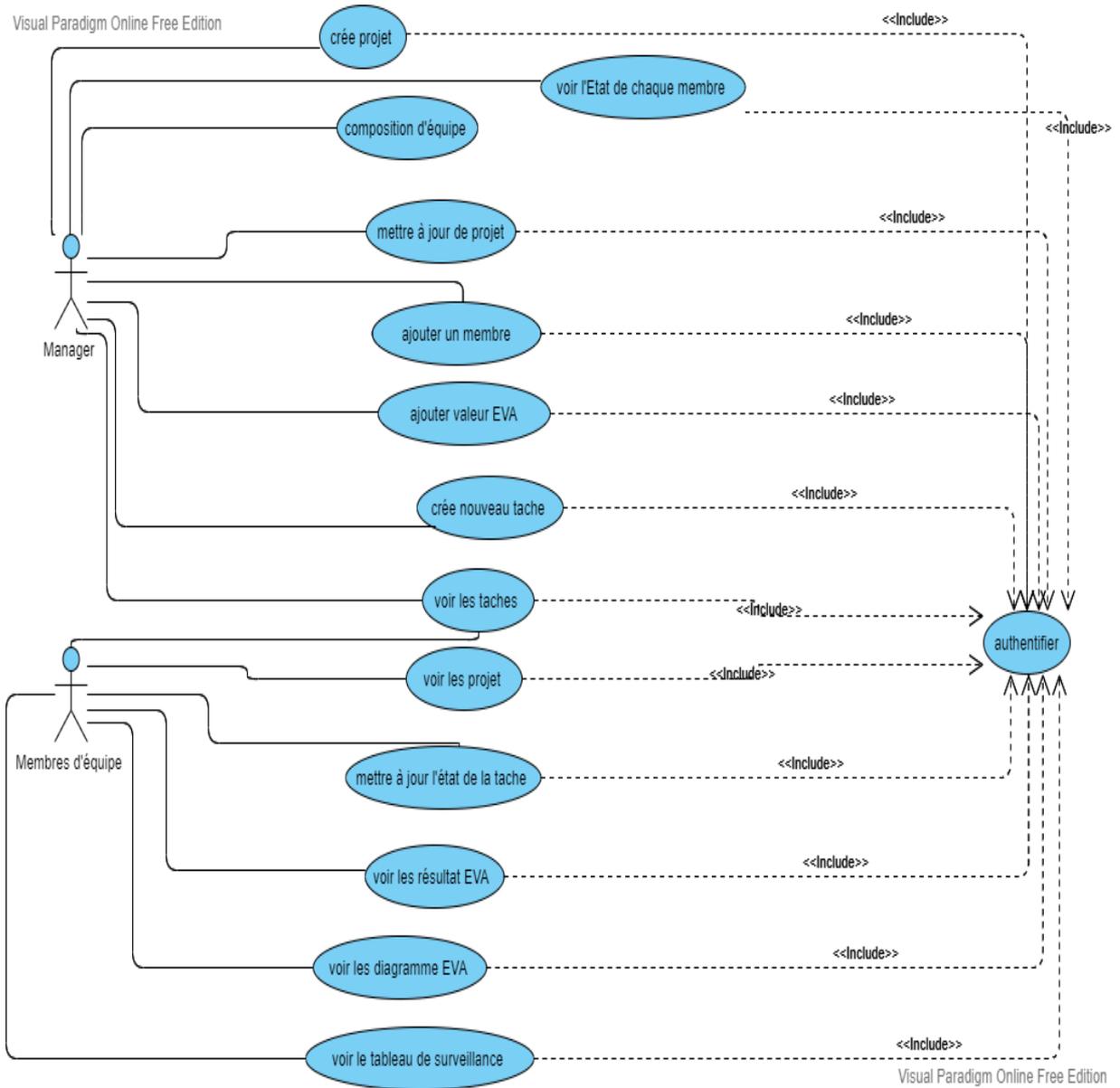


Figure 12 – Diagramme de cas d'utilisation

3.5 Analyse logique

Ce cycle consiste à créer un modèle d'analyse du système qui définit la structure et le comportement des objets connus dans le métier des acteurs dans le cadre de la mise en application de leurs besoins. Celui-ci commence par l'analyse des scénarisons fonctionnels et finit par le développement de modèle des catégories finales.

Le diagramme de classe suivant décrit les structures d'objets et d'informations utilisées par notre application, à la fois en interne et dans la communication avec ses utilisateurs.

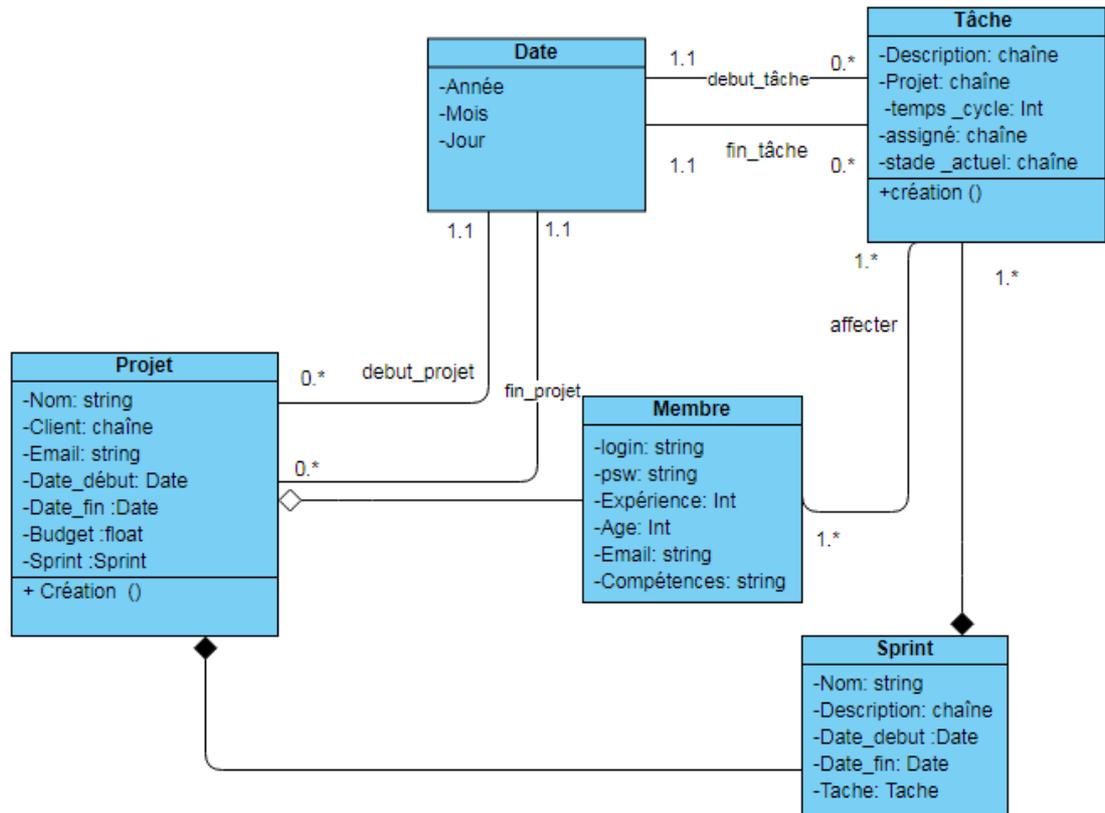


Figure 13 – Diagramme de classe

Nous détaillons dans ce qui suit le modèle dynamique qui permet d'examiner le comportement des objets, et les modifications d'états des objets suite aux réceptions de messages. Ci-après le diagramme de séquence qui permet de décrire comment les éléments du système interagissent entre eux et avec les acteurs.

a- L'authentification

L'utilisateur (membre, administrateur) a la possibilité de s'authentifier en saisissant un Email et un mot de passe dans les champs qui leurs sont réservés. Après validation, l'utilisateur accède à la session qui lui est propre. Si les données entrées sont correctes il se connecte, sinon le système renvoi un message d'erreur (voir Figure 14).

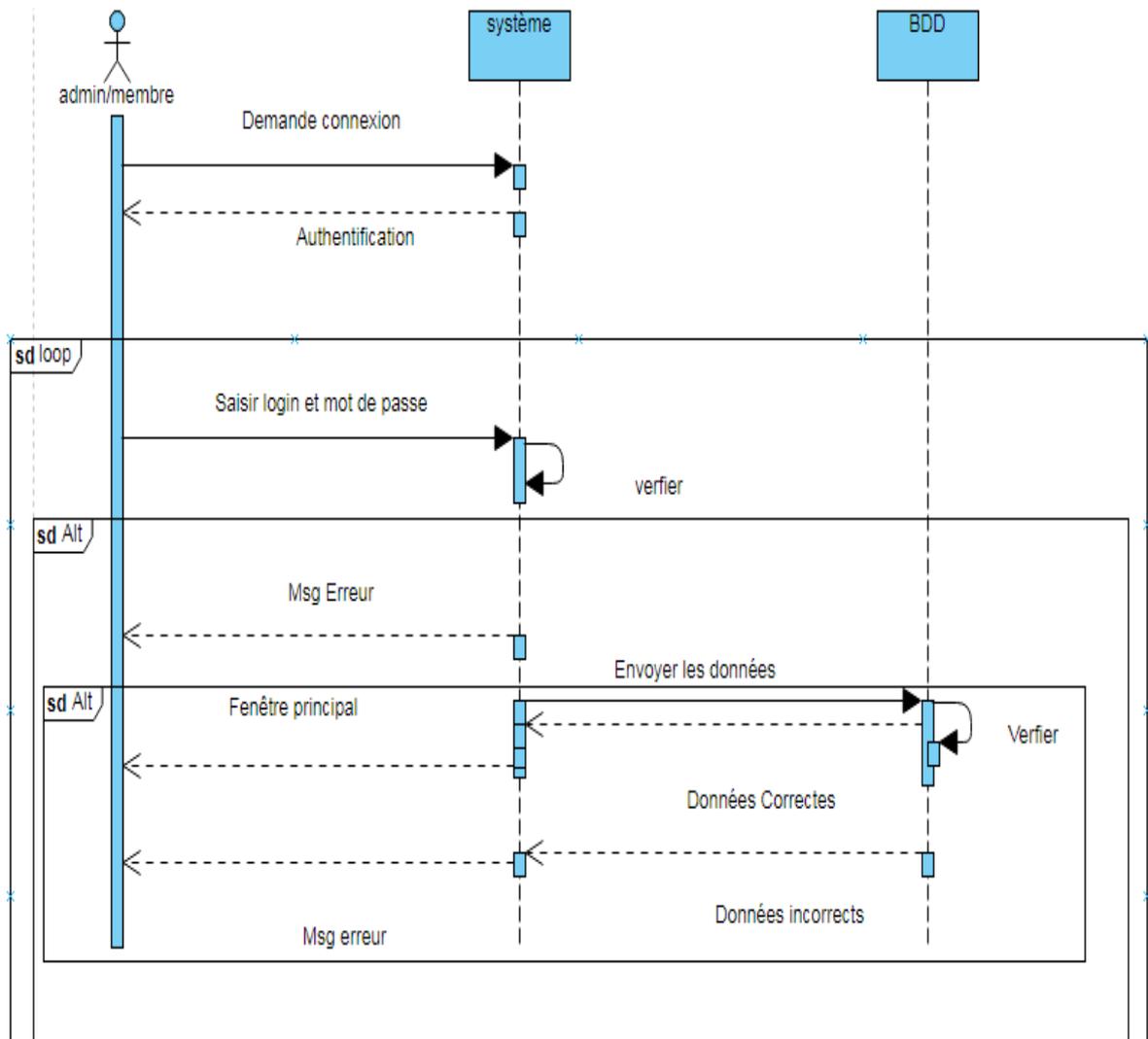


Figure 14 – Diagramme de séquence du système pour l’authentification.

b- Création de projet

L'utilisateur peut dans ce cas de figure avoir la possibilité de créer, modifier et supprimer un projet.

Dans la création d'un nouveau projet, le système lui propose un formulaire à remplir. Si aucune erreur n'est commise alors le système envoie une requête d'ajout à la BDD et là confirme, sinon il affiche un message d'erreur. L'utilisateur peut aussi modifier ou supprimer un projet sélectionné (voir Figure15).

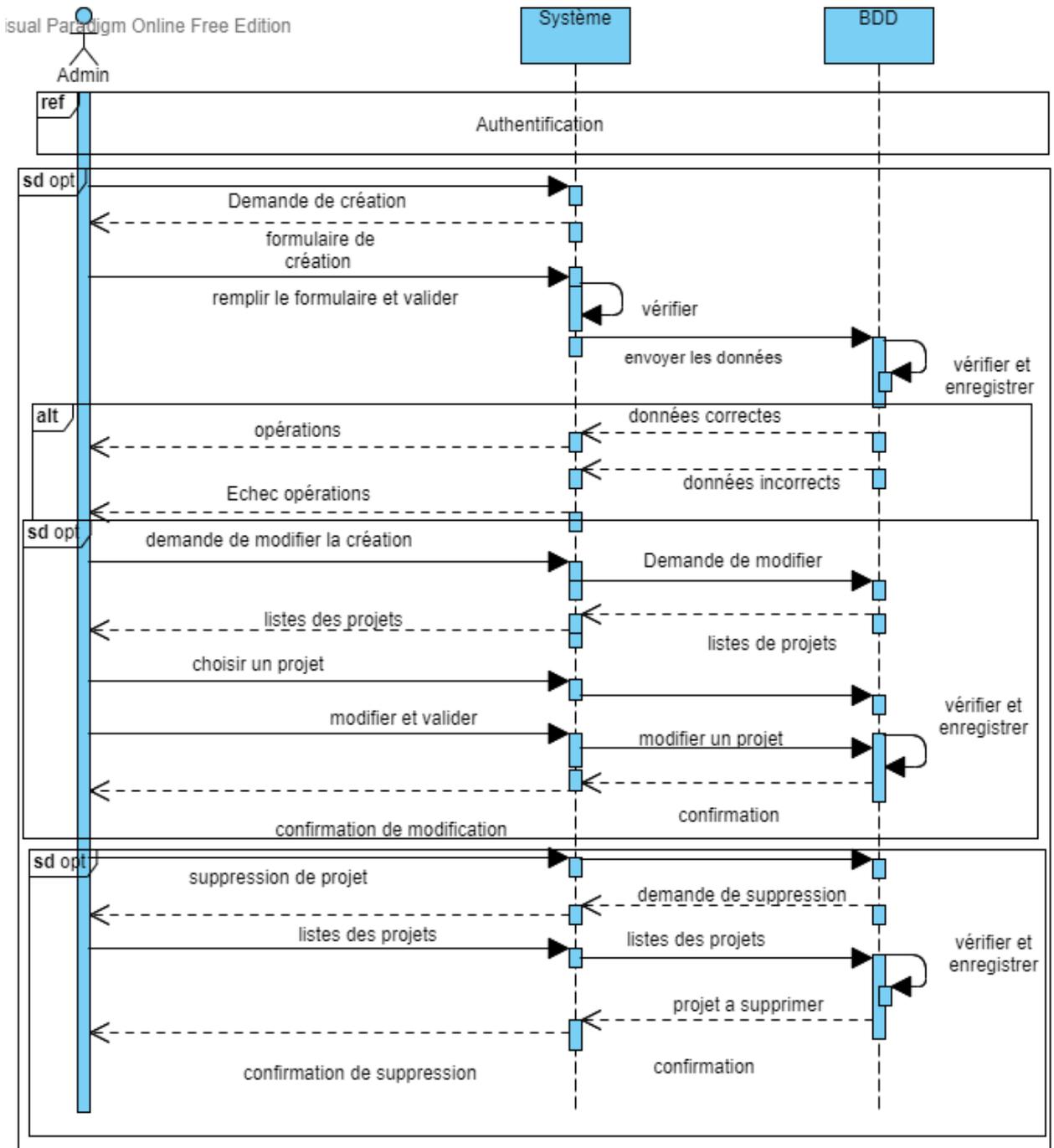


Figure 15 – Diagramme de séquence du système « création de projet ».

c- Inscription

Afin de s’inscrire, l'utilisateur doit d'abord s'authentifier ensuite répondre au formulaire. Le système vérifie la validité des champs et l’identité de l'utilisateur. Si l'inscription n'est pas valide, l'utilisateur doit réinscrire puis faire le test sur les traits de personnalité s’il s’agit d’un membre (voir Figure 16).

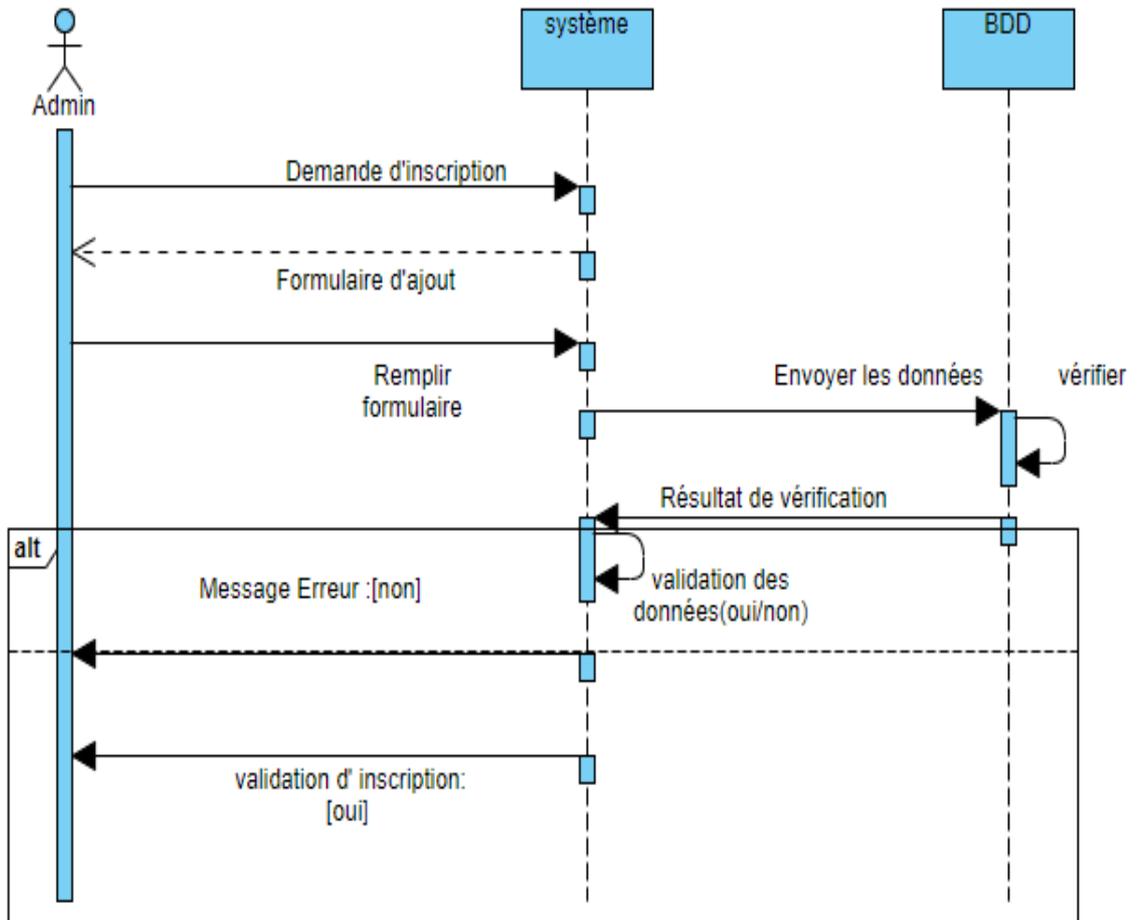


Figure 16 – Diagramme de séquence du système « Inscription un membre ».

d- Ajout d'une tâche

Afin d'ajouter des tâches, l'utilisateur doit d'abord être authentifié. Il remplit le formulaire d'ajout d'une tâche en renseignant les informations nécessaires. En ajoutant une tâche, le système doit vérifier si la saisie est correcte ainsi que l'inexistence préalable de la tâche afin de valider son ajout (voir Figure 17).

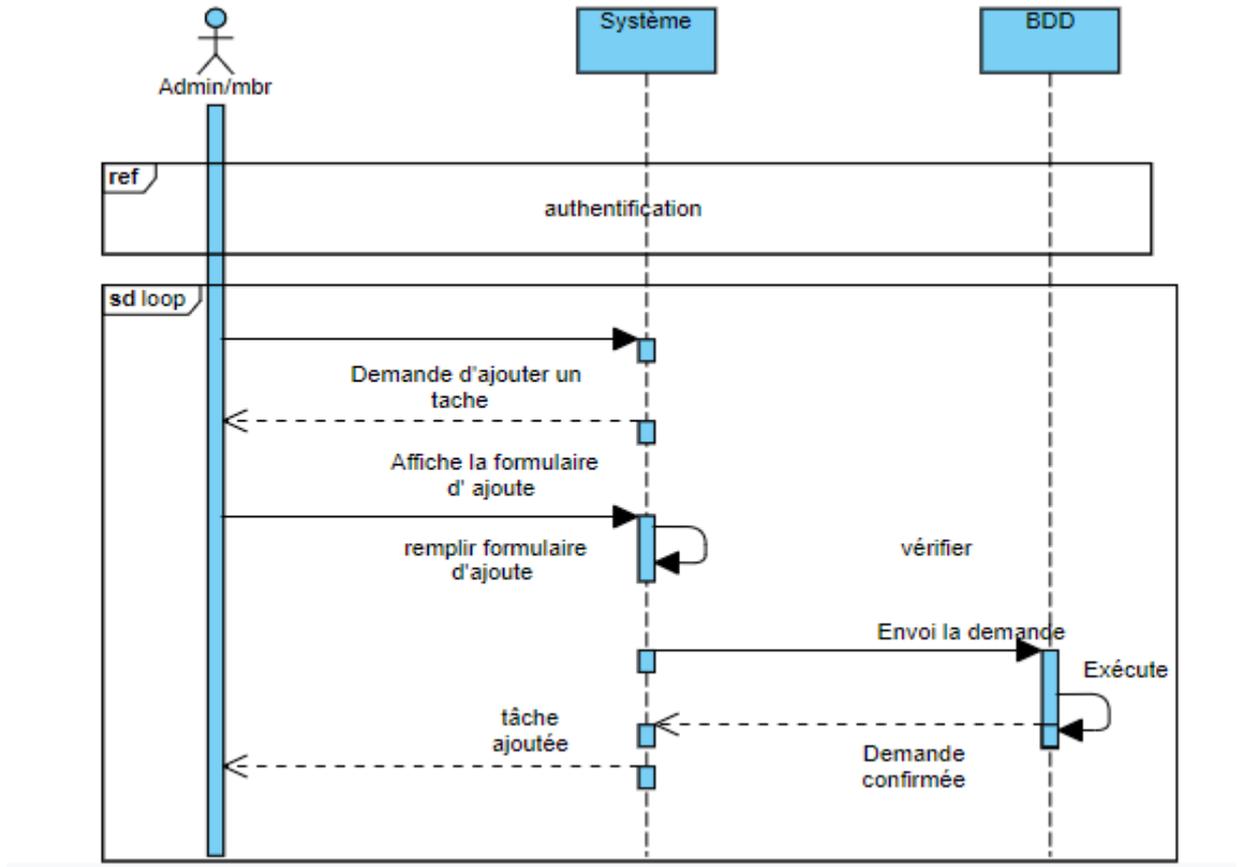


Figure 17 – Diagramme de séquence du système « Ajouter une Tâche ».

3.6 Conclusion

Dans ce chapitre nous avons présenté d'une façon globale, les deux étapes essentielles de l'analyse et la conception de notre application. Le chapitre suivant sera consacré à la description de notre contribution de projet de Master ainsi que la phase de développement et de test de notre application.

Chapitre IV: IMPLEMENTATION

4.1 Introduction

Après avoir présenté la conception qui répondait aux besoins étudiés, nous entamons la partie réalisation de l'application. Nous expliquons les différents outils et langages de développement utilisés ainsi que les détails de l'implémentation de notre application qui inclut la base de données et un aperçu sur les interfaces utilisateur.

4.2 Description de l'environnement de développement

4.2.1 Java entreprise édition

Le terme « Java EE » anciennement raccourci en « J2EE » signifie Java Enterprise Edition, il fait référence à une extension de la plate-forme standard. Autrement dit, la Plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE à quoi s'ajoute un grand nombre de bibliothèques remplissant différentes fonctionnalités que la plate-forme standard ne remplit pas d'origine. L'objectif majeur de Java EE est de faciliter le développement d'applications Web robustes et distribuées, déployées et exécutées sur un serveur d'applications.

Java EE regroupe un ensemble d'API chacun est destiné à faire un traitement spécifique. Dans le cadre de notre application, nous avons choisi d'utiliser les API suivants [47] :

- Servlet : Une servlet est une classe Java, qui a la particularité de permettre le traitement de requêtes et la personnalisation de réponses.
- JSP (Java Server Pages) : Les pages JSP sont une des technologies de la plate-forme Java EE les plus puissantes, simples à utiliser et à mettre en place. Elles se présentent sous la forme d'un simple fichier au format texte, contenant des balises respectant une syntaxe à part entière. Le langage JSP combine à la fois les technologies HTML, XML, Servlet et JavaBeans en une seule solution permettant aux développeurs de créer des vues dynamiques.

IMPLEMENTATION

4.2.2 Le serveur Apache-Tomcat

Apache-Tomcat est le serveur d'application Java du projet Jakarta de la fondation Apache. Ce serveur libre, sous licence Apache permet d'exécuter des applications web développées avec les technologies java (Servlet, JSP . . .) [47].

4.2.3 MySQL (My Structured Query Language)

Est une base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web), basé sur un modèle client - serveur. Son rôle consiste à stocker et à gérer une grande quantité de données en les organisant sous forme de tables. MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment) [49].

4.2.4 CSS

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML [47].

4.2.5 JavaScript

Le JavaScript est un langage informatique utilisé sur les pages web. Le langage a été créé en 1995 par Brendan Eich pour le compte de Netscape Communications Corporation, Ce langage à la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activés côté serveur. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer.

La particularité du JavaScript consiste à créer des petits scripts sur une page HTML dans le but d'ajouter une petite animation ou un effet particulier sur la page. Cela permet en général d'améliorer l'ergonomie ou l'interface utilisateur, mais certains scripts sont peu utile et servent surtout à ajouter un effet esthétique à la page. L'intérêt du JavaScript est d'exécuté un code sans avoir à recharger une nouvelle fois la page [47].

4.2.6 L'IDE Eclipse

Un IDE (environnement de développement intégré) est un outil qui a pour objectif de faciliter le développement sous un ensemble restreint de langages. Il contient un ensemble de

IMPLEMENTATION

fonctionnalités permettant de faciliter la tâche sous programmeur [47]. Nous avons utilisé l'IDE Eclipse qui est libre, extensible et polyvalent développé en JAVA. Il permet de créer des projets de développement sous n'importe quel langage de programmation. Eclipse est principalement développé autour de la notion de plug-in. Ceci permet à ses utilisateurs de l'adapter selon leurs besoins.

4.3 Tableau de bord PMT

Cette partie présente les interfaces PMT ainsi que leurs fonctions et caractéristiques. En particulier, l'explication est alignée sur la démonstration des fonctions des composants PMT.

Le PMT commence par la page de connexion où le nom d'utilisateur et le mot de passe sont saisis comme indiqué sur la figure 18. Les utilisateurs de PMT peuvent être des chefs de projet ou / et des membres de l'équipe qui travaillent sur le développement de projets logiciels. Chaque utilisateur dispose de son propre code d'authentification pour accéder au système. Le PMT donne à chaque membre de l'équipe un nom d'utilisateur et un mot de passe spécifiques.

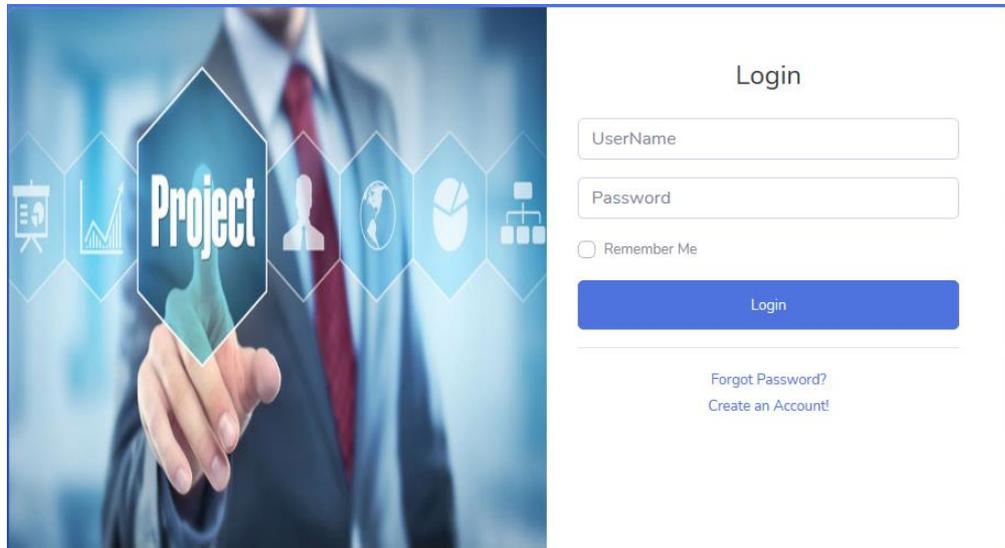


Figure 18 -Page de connexion

IMPLEMENTATION

Une fois la connexion réussie, la page d'accueil apparaît et affiche un aperçu du PMT comme illustré à la Figure 19.

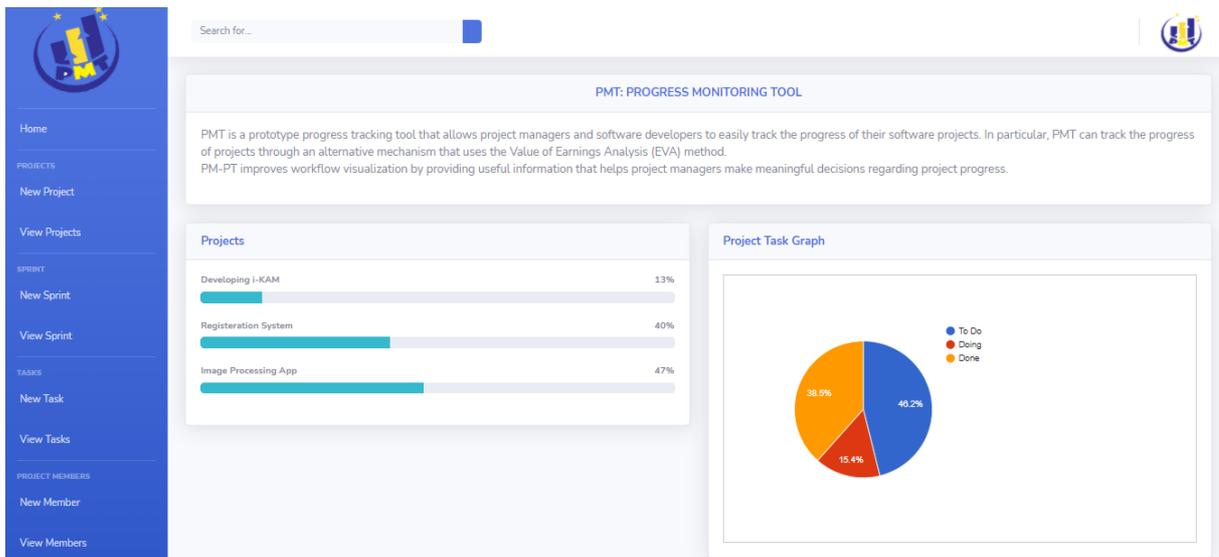


Figure 19 - Page d'accueil

4.4 Création de projets, sprint et de tâches dans PMT

Sous l'onglet PROJET, le Manager peut créer un nouveau projet en saisissant les informations principales sur le projet, puis en cliquant sur le bouton Enregistrer comme indiqué sur la figure 20. Les informations nécessaires incluent le nom du projet, le nom du client, le numéro de contact, l'email, les dates de début et d'achèvement et le budget total du projet. Ces informations sont stockées dans la base de données pour être utilisées pendant le processus de développement du projet.

New Project

Project Name:

Customer:

Email:

Start Date:

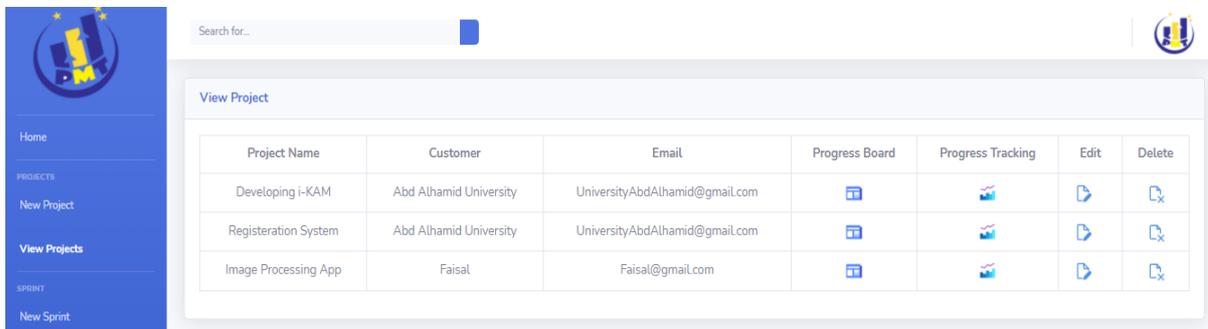
Complete Date:

Budget At Completion:

Figure 20 - Ajouter un nouveau projet

IMPLEMENTATION

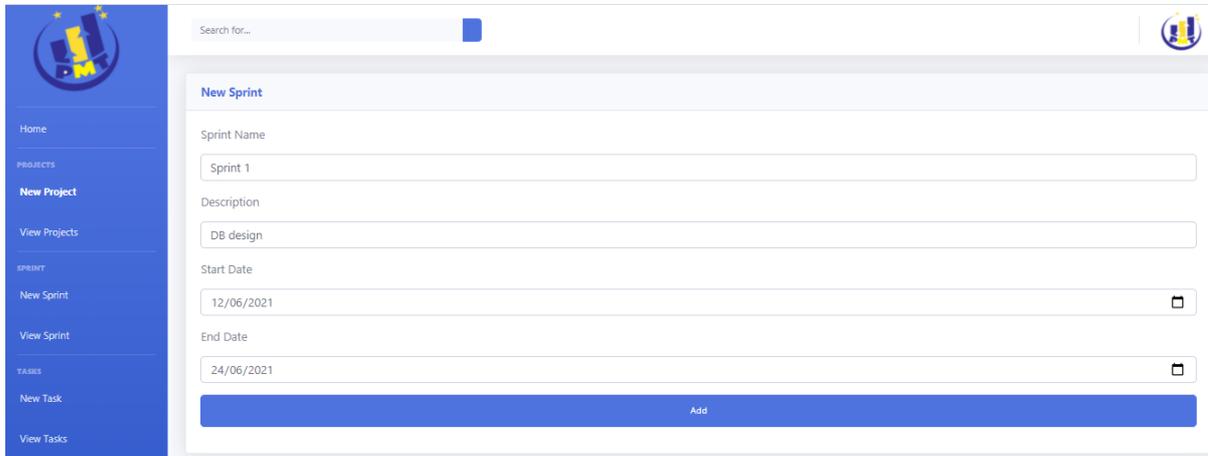
Les projets existants peuvent être visualisés en cliquant sur « Afficher les projets » sous l'onglet « PROJET », comme illustré à la Figure 21.



Project Name	Customer	Email	Progress Board	Progress Tracking	Edit	Delete
Developing i-KAM	Abd Alhamid University	UniversityAbdAlhamid@gmail.com				
Registration System	Abd Alhamid University	UniversityAbdAlhamid@gmail.com				
Image Processing App	Faisal	Faisal@gmail.com				

Figure 21- Afficher la liste des projets

Sous l'onglet SPRINT, le Manager peut créer un des sprints associés au projet. Les informations nécessaires incluent la description, la date de début et la date de fin. La figure 22 montre un exemple de formulaire de création d'un sprint.



New Sprint

Sprint Name:

Description:

Start Date:

End Date:

Figure 22- Ajouter un nouveau Sprint.

Sous l'onglet TÂCHES, des tâches de projet peuvent être créées qui sont associées à des projets. La figure 23 représente la page de création d'une nouvelle tâche en saisissant la description de la tâche et en identifiant le projet auquel elle appartient, ainsi que le temps de cycle (en jours) pendant lequel la tâche doit être exécutée. Les informations saisies sont également stockées dans la base de données après avoir cliqué sur le bouton Enregistrer.

IMPLEMENTATION

The screenshot shows a 'New Task' form with the following fields:

- Task Description:
- Project:
- Cycle Time (In Days):
- Assigned to:
- Current Stage:

A blue 'Save' button is located at the bottom of the form.

Figure 23- Ajouter une nouvelle tâche

De plus, les tâches peuvent être visualisées en sélectionnant le nom du projet dans la liste déroulante, comme illustré à la figure 24. Cette page montre la description de toutes les tâches, les membres de l'équipe qui ont été assignés à les effectuer et l'étape actuelle de la tâche à accomplir. Dans cette page, les tâches peuvent être mises à jour en cliquant sur l'icône d'édition, grâce à laquelle différents critères de tâches peuvent être mis à jour. Les critères incluent la description de la tâche, le membre de l'équipe et l'étape actuelle.

The screenshot shows a 'View Project Tasks' page with a dropdown menu for 'Select the project' set to 'Registration System'. Below is a table of tasks:

Task Description	Assigned to	Current Stage	Edit	Delete
Student details page	Hamzah	to do		
DB design	Yaseen	doing		
About page	Abubakr	done		
Schools page	Salim	to do		
Logo design	Ali	done		
Colleges page	Mohsen	done		
Student details page	Mohsen	failed		
DB design	Ali	failed		
About page	Salim	failed		

Figure 24- Afficher les tâches des projets

IMPLEMENTATION

4.5 Suivi de progression de projet dans PMT

Nous avons intégré à notre outil la méthode de suivi de progression de projet EVA. Elle comporte quatre indicateurs importants, à savoir :

- la variance des horaires (SV)
- l'indice de performance des horaires (SPI),
- la variance des coûts (CV)
- l'indice de performance des coûts (IPC).

Ces indicateurs permettent de suivre l'avancement du projet. Ils sont générés en calculant quatre équations, qui sont:

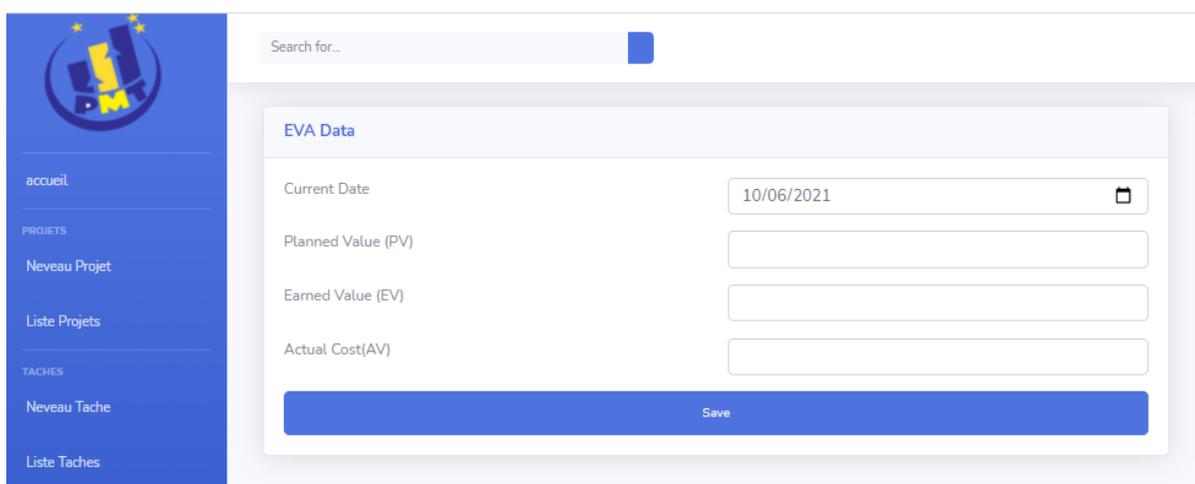
$$SV = EV - PV \quad (1)$$

$$SPI = EV/PV \quad (2)$$

$$CV = EV - AC \quad (3)$$

$$CPI = EV/AC \quad (4)$$

Pour suivre l'avancement du projet à l'aide d'EVA, cliquez sur l'icône correspondante, puis saisissez les trois valeurs à savoir la valeur planifiée (PV), le coût réel (AC) et la valeur acquise (EV). Cliquez ensuite sur le bouton Enregistrer par lequel la date actuelle est automatiquement capturée comme illustré à la Figure 25.



The screenshot shows a web application interface for entering EVA data. On the left is a blue sidebar with navigation links: 'accueil', 'PROJETS' (with sub-links 'Neveau Projet' and 'Liste Projets'), and 'TACHES' (with sub-links 'Neveau Tache' and 'Liste Taches'). The main content area has a search bar at the top. Below it is a form titled 'EVA Data' with four input fields: 'Current Date' (pre-filled with '10/06/2021' and a calendar icon), 'Planned Value (PV)', 'Earned Value (EV)', and 'Actual Cost(AV)'. A blue 'Save' button is at the bottom of the form.

Figure 25- Données EVA saisies

IMPLEMENTATION

Pour un suivi progression efficace du projet, nous avons intégré plusieurs graphiques permettent au gestionnaire et à son équipe de mesurer facilement les extrants et les résultats obtenus, de planifier les tâches et de respecter les délais comme illustré dans les figures ci-dessous.

La première barre permet au manager de savoir l'avancement du Sprint en cours. Elle montre le temps exercé dans le Sprint. Les trois autres barres montrent les tâches à faire, les tâches en cours, les tâches effectuées et tâches annulées ou échoués. Le tableau facilite au manager de savoir quel sont les tâches qui ne sont pas encore faites par chaque membre.

Dans le cadre du volet de visualisation du Workflow, les résultats d'EVA sont visualisés sous forme de diagrammes afin de faciliter l'analyse du suivi de l'avancement. Qui présente le flux des écarts de coûts et d'échéanciers selon les périodes de temps.

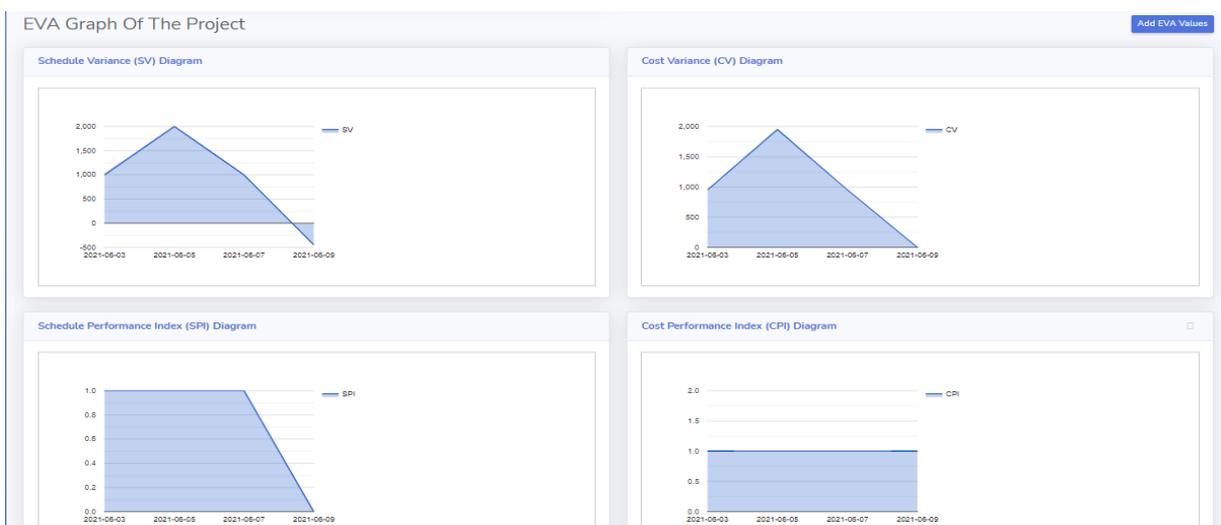
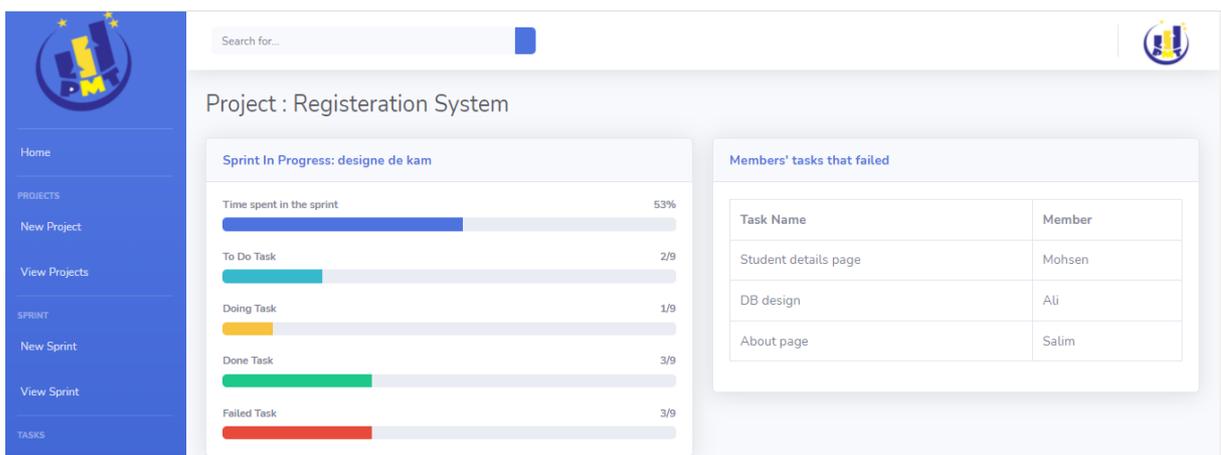


Figure 26- Le tableau de suivi des progrès amélioré

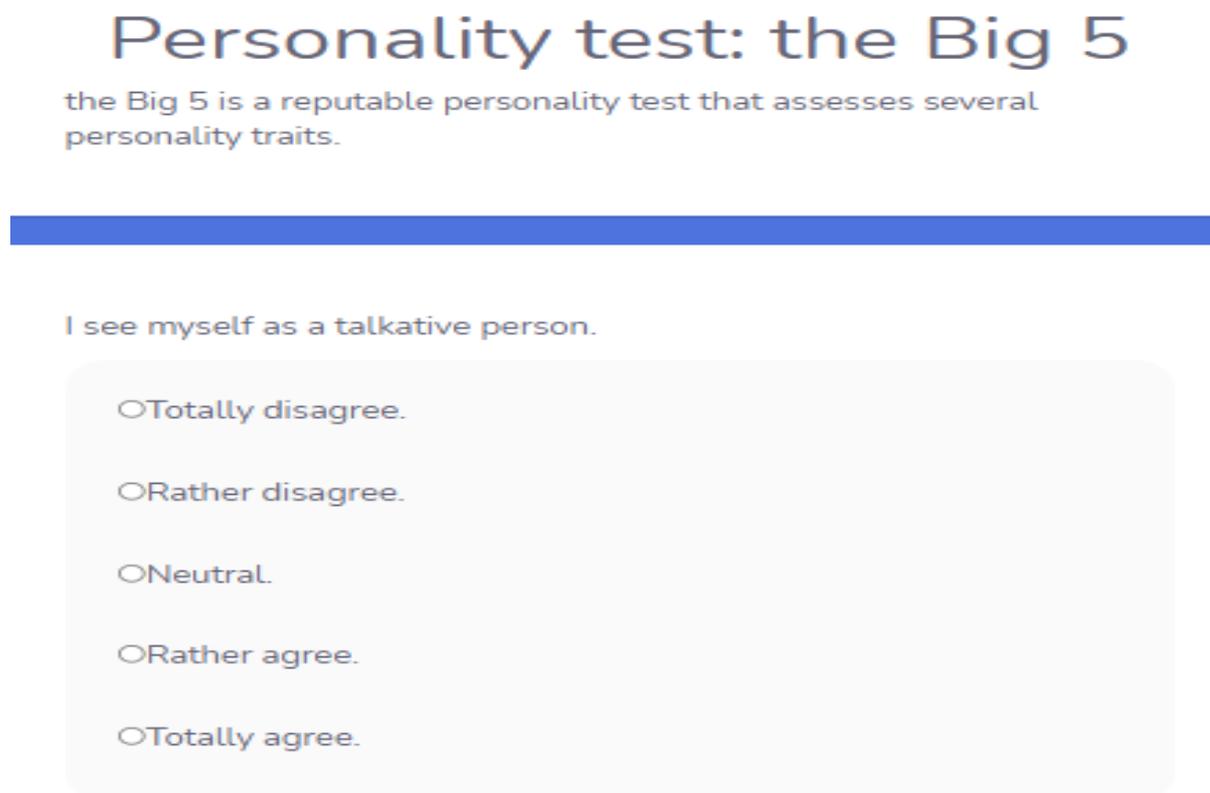
IMPLEMENTATION

4.6 Aide à la composition d'équipe dans PMT

Quand un membre fait une inscription, il doit passer un test personnalité (voir figure 27). En effet, les tests de personnalité sont de plus en plus inclus dans les entretiens d'embauche afin de vérifier si un candidat peut répondre à certaines exigences du poste. Dans notre outil PMT, le résultat de ce test est présenté afin de donner au gestionnaire de projet les informations supplémentaires sur les membres qui composent les équipes de son projet.

Le gestionnaire pourra ainsi trouver une meilleure assistance pour composer des équipes qui collaborent efficacement. Par exemple, selon son trait de personnalité extravertie ou intravertie, le gestionnaire peut comprendre les préférences des membres en termes de travail sur des tâches collaboratives. Selon leurs personnalités, certains membres peuvent être plus efficaces en travaillant en toute autonomie, et d'autres, à l'inverse, seront plus productifs en équipe. Cela aide donc le gestionnaire à sélectionner le membre convenable.

La composition d'équipe ne se limite pas à la personnalité, d'autres paramètres sont montrés au gestionnaire notamment : la disponibilité des membres en temps réel, leurs compétences, leurs historique de collaboration, leurs anciennes performances, etc.



Personality test: the Big 5
the Big 5 is a reputable personality test that assesses several personality traits.

I see myself as a talkative person.

- Totally disagree.
- Rather disagree.
- Neutral.
- Rather agree.
- Totally agree.

Figure 27- Test de personnalité des membres

IMPLEMENTATION

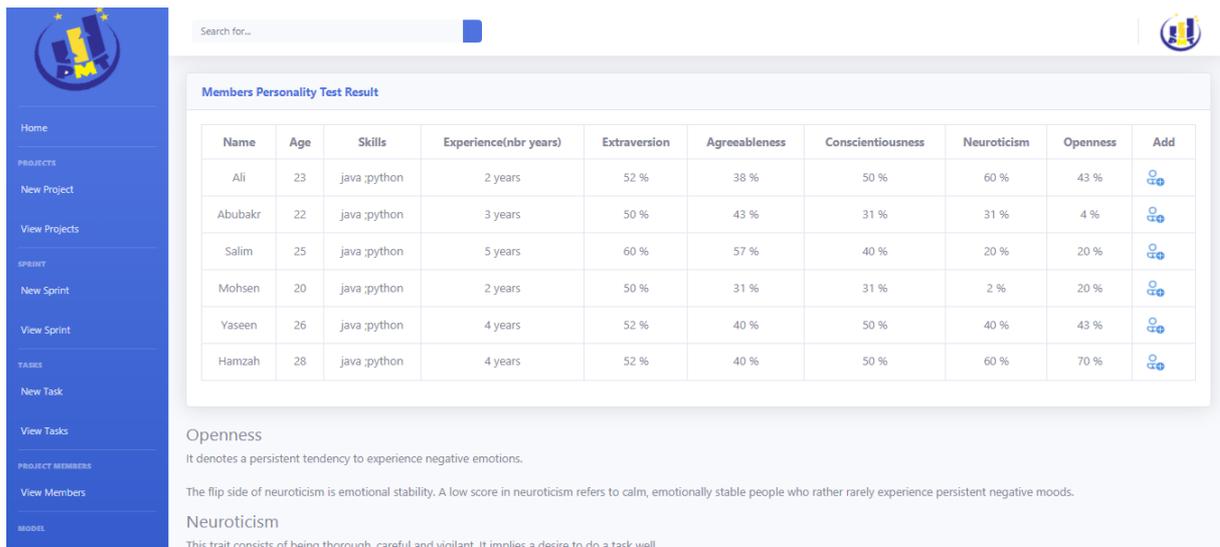


Figure 28- Composition d'équipe

4.7 Conclusion

Dans ce dernier chapitre nous avons illustré les détails de la réalisation de notre prototype de gestion de progression de projet. Nous avons pu mettre en œuvre à travers ses différentes interfaces les métriques et les tests nécessaires à la compréhension de l'avancement d'un projet ainsi que les membres qu'il le compose. L'objectif est de permettre au gestionnaire d'avoir suffisamment d'information sur le projet, planifier/modifier des objectifs et former des équipes efficaces lors d'un projet Agile.

Cet outil proposé est encore en phase de prototypage et ne représente pas l'outil final. Vu les contraintes temporelles de ce projet de Master, nous nous sommes limité au choix le plus simple en termes de maîtrise de la technologie, le langage de programmation ainsi qu'aux fonctionnalités intégrées. Ce travail de Master ouvre des perspectives pour améliorer les fonctionnalités de l'outil et de le tester afin d'avoir les retours des gestionnaires de projets de développement logiciel.

CONCLUSION GENERALE

CONCLUSION GENERALE

L'utilisation des outils SPM permet de faciliter la gestion de projets Agile et améliorer l'efficacité de gestion qui affecte par conséquent la qualité du projet. Dans ce rapport de projet Master, nous avons présenté d'abord notre résultat de recherche bibliographique, et analyse de l'état de l'art des outils SPM utilisés pour l'aide au suivi de progression de projet et de composition des équipes performantes. Nous avons présenté les principaux outils SPM existants en analysant leurs réponses à nos critères d'analyse. Nous avons trouvé que la plupart des outils récents adoptent les principes de la méthode Scrum et Kanban qui consistent à visualiser le workflow et à contrôler le work in progress (WIP). Cependant, la plupart de ces outils souffrent de mécanismes de suivi des progrès insuffisants. De plus, les outils existants ne fournissent pas de l'assistance en termes de composition d'équipe.

Suite à cette analyse de l'état de l'art, nous avons déterminé notre contribution par rapport à notre problématique de recherche de ce projet de Master. Il s'agit de développer un outil SPM alternatif en améliorant les fonctionnalités actuelles et en surmontant les obstacles rencontrés lors de l'utilisation des outils existants. Ceci aidera le Manager à construire un travail d'équipe efficace. Non seulement les expériences et les compétences des membres de l'équipe à prendre en compte, mais aussi leurs traits de personnalité. L'outil aide également le gestionnaire à obtenir et à suivre l'avancement du projet grâce à un mécanisme de suivi avec la possibilité de visualiser des informations cruciales.

BIBLIOGRAPHIE

1. PMI. Pulse of Profession. 2017, 26 July 2019. <https://www.pmi.org/>
2. Mishra, D., A. Mishra. Complex Software Project Development: Agile Methods Adoption. *Journal of Software Maintenance and Evolution: Research and Practice*, Vol. 23, 2011, No 8, pp. 549-564.
3. Beck, K., et al. Manifesto for Agile Software Development. 2001. <https://Agilemanifesto.org/>
4. Lin, J., et al. Studying Task Allocation Decisions of Novice Agile Teams with Data from Agile Project Management Tools. – In: Proc. of 29th ACM/IEEE International Conference on Automated Software Engineering, ACM, 2014.
5. H. S. Packer, A. Chapman, and L. Carr, “GitHub2PROV: Provenance for Supporting Software Project Management,” in *11th International Workshop on Theory and Practice of Provenance (TaPP 2019)*, New York, NY, USA, 2019.
6. H. Alaidaros and M. Omar, “Software Project Management Approaches for Monitoring Work-In-Progress: A Review,” *Journal of Engineering and Applied sSciences*, vol. 12, pp. 3851-3857, 2017.
7. S. Ambler, *2018 IT project success rates survey results*, 2018. Accessed on : Jun. 19, 2019. [Online]. Available: <http://www.ambysoft.com/surveys/success2018.html#Results>
8. The Standish Group, *Chaos Report: The Winning Hand*, 2015. Accessed on: Jul. 11, 2019. [Online]. Available: <http://www.standishgroup.com/store/>
9. Gestion de projet stratégique : Équipe de projet : [https://fr.wikiversity.org/wiki/Gestion de projet stratégique équipe de projet](https://fr.wikiversity.org/wiki/Gestion_de_projet_strat%C3%A9gique_%C3%A9quipe_de_projet)
10. Caitlin McCormack , Introduction à l'agilité : <https://monday.com/blog/project-management/introduction-to-Agile/>
11. Beck, K., et al. Manifesto for Agile Software Development. 2001. <https://Agilemanifesto.org/>
12. The top project management methodologies: which is right for you? <https://monday.com/blog/project-management/top-project-management-methodologies/>
13. Judicaël PAQUET, Dominique POPIOLEK-OLLÉ, Mai Ly (coach Agile), *Sprint scrum–les cérémonies*, 2020 : <https://blog.myAgilepartner.fr/index.php/2018/08/17/ceremonies-sprint-scrum/>
14. LA MÉTHODE KANBAN POUR L'IT : PRINCIPES, FONCTIONNEMENT, AVANTAGES ET COMPARAISON: <https://www.apollo-formation.com/gestion-de->

BIBLIOGRAPHIE

- [projet/methode-kanban-it.html](#)
15. DAVID GALIANA, 14 KPI à connaître pour évaluer le suivi d'un projet, 2019. [HTTPS://WWW.WIMI-TEAMWORK.COM/FR/BLOG/KPI-EVALUER-SUIVI-PROJET/](https://www.wimi-teamwork.com/fr/blog/kpi-evaluer-suivi-projet/)
 16. Onanong Hlaoittinun, Eric Bonjour, Maryvonne Dulmet. Méthodologie pour la constitution d'équipes de projet de conception.. 7ème Congrès International de Génie Industriel, GI'2007, Trois Rivières., Jun 2007, Québec, Canada. sur CD ROM - 10 p. fihal-00162960.
 17. M. Zulkefli, Y. Saadiah, and A. Noor Habibah, "Success Determinants in Agile Software Development Methodology," *Procedia Eng.*, no. Icmlc, pp. 203–207, 2011.
 18. J. F. Salgado, V. Chockalingam, and D. S. Ones, "Predictors used for personnel selection: An overview of constructs, methods and techniques: Personnel psychology," in *Handbook of industrial, work and organizational psychology: Volume 1 Personnel Psychology*, N. Anderson, D. S. Ones, H. K. Sinangil, and C. Viswesvaran, Eds. London: SAGE Publications, 2005, pp. 165–199.
 19. D. H. Prasad and M. H. Vasanth, "Value of Bottom-up Team Formation for Complex Adaptive Business Systems," in *Systems Conference (SysCon), 2014 8th Annual IEEE*, 2014, pp. 272–276.
 20. S. C. Wong, S. Selvadurai, V. Saxena, M. Abdul, and K. Okal, "Demographic Diversity and Salience of Nationality on Team Effectiveness in Information and Communication Technology (Ict) Small and Medium Enterprises (Smes)," *J. Soc. Sci. Humanit.*, vol. 12, no. 2, pp. 229–244, 2017.
 21. Project Management Institute, A guide to the project management body of knowledge (PMBOK ® guide), 6th ed. 2017.
 22. A. R. Gilal, J. Jafreezal, B. Shuib, O. Mazni, and A. Abro, "Impact of Software Team Composition Methodology on The Personality Preferences of Malaysian Students," in *3rd International Conference*, 2016, no. August, pp. 454–458.
 23. M. R. Mawarny, J. Noble, and S. Marshall, "Agile self-selecting teams foster expertise coordination," *Interdiscip. J. Information, Knowledge, Manag.*, vol. 14, 2019.
 24. K. Lee and M. C. Ashton, "Psychometric Properties of the HEXACO-100," *Assessment*, 2016.
 25. Jean Claude GROSJEAN, Indicateurs KPI pour nos Equipes Agiles , 2007,<https://www.qualitystreet.fr/2017/11/13/kpi-Agiles/>
 26. Sinem Besir, Kökten Ulas Birant, A Case for Decision Support Systems on Project

BIBLIOGRAPHIE

- Management, Dokuz Eylül University, Computer Engineering Department, Buca, Izmir, Turkey, 2010.
27. Atlassian, *Move work forward*, 2019. Accessed on: Jul. 20, 2019. [Online]. Available: <https://www.atlassian.com/>
 28. Everhour Blog, *14 best project management tools – pros and cons*, 2018. Accessed on: Jul. 18, 2019. [Online]. Available: <https://everhour.com/blog/14-best-project-managementtools/>
 29. J. K. Bajwa and J. Kaur, “Comparative study of APM tools, ” *International Journal of Engineering Science*, vol. 24, pp. 26-35, 2017.
 30. V. Abramova, F. Pires, and J. Bernardino, “Open source and proprietary project management tools for SMEs, ” *Journal of Information Systems Engineering & Management*, vol. 1, pp. 177- 186, 2016.
 31. Deniz Özkan¹, Alok Mishra , Agile Project Management Tools: A Brief Comparative View , 2019
 32. Estelle Raffin, monday.com : le meilleur outil pour gérer vos projets d'équipe à distance, 2020. <https://www.blogdumoderateur.com/monday-com-meilleur-outil-gestion-projets-equipe-a-distance/>
 33. Basecamp, *Get it together and manage projects the right way*, 2019. Accessed on: Jul. 21, 2019. [Online]. Available: <https://www.basecamp.com/>
 34. Everhour Blog, *14 best project management tools – pros and cons*, 2018. Accessed on: Jul. 18, 2019. [Online]. Available: <https://everhour.com/blog/14-best-project-managementtools/>
 35. S. Sawant, “Collaborative online learning tools and types: Few perspectives of its use, ” in *Human Development and Interaction in the Age of Ubiquitous Technology*, ed USA: IGI Global, 2016, pp. 94-119.
 36. Pavić, *Redmine Cookbook*, 1st ed. United Kingdom: Packt Publishing Ltd, 2016.
 37. M. S. Iqbal, M. Sidaq, M. Shabbir, A. Rehman, M. Sajad, and T. Khan, “Assessment of inspection tools with standard, management review, technical review, inspection and walkthroughs, ” *International Journal of Computer Science and Information Security*, vol. 14, pp. 622-633, 2016.
 38. V. Abramova, F. Pires, and J. Bernardino, “Open source and proprietary project management tools for SMEs, ” *Journal of Information Systems Engineering & Management*, vol. 1, pp. 177- 186, 2016.
 39. Hamzah Alaidaros, Mazni Omar, Rohaida Romli and Adnan Hussein. The

BIBLIOGRAPHIE

- Development and Evaluation of A Progress Monitoring Prototype Tool for Software Project Management. International Conference of Intelligent Computing and Engineering, 2019.
40. Boehm, BW, Tourneur, R, Agilité d'équilibrage et Discipline: un guide fou le Perplexe. Addison-Wesley, lecture (2003).
 41. Kraut, RE, Streeter, LA: Coordination en développement logicielment. Communications de laACM 38, 69–81 (1995).
 42. Hoegl, M., Gemuenden, H.G, Travail d'équipe Qualité et le Succès of Innovant Projets: Un concept théorique et des preuves empiriques. Organisation science 12, 435–449 (2001)
 43. Martine Fowler, BRIEF GUIDE TO THE STANDARD OBJECT MODELING LANGUAGE, UML dislilled third edition, 2003.
 44. Michel Winter, MÉTHODE DE DÉVELOPPEMENT « RUP », 2012.
 45. MAGALI Contensin, Bases de données et Internet avec PHP et MySQL, DUNOD, 2004,1 ère édition.
 46. Pascal ANDRE, Développement de logiciel avec UML, Université de Nantes.
 47. BENACHOUR Kouceila, Conception et réalisation d'une application Web E-Commerce sous Java EE, 2015.
 49. Magali Contensin, Bases de données et Internet avec PHP et MySQL , éditions Dunod, 2004.