

**UQAC**

Université du Québec  
à Chicoutimi

**EXPLOITATION DE L'ANALYSE AUTOMATIQUE  
DU LANGAGE POUR AMÉLIORER LA GESTION  
DE PROJET AVEC LE CADRE SCRUM**

**PAR GAEL TCHAWÉ DISSI**

**MÉMOIRE PRÉSENTÉ  
À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI  
EN VUE DE L'OBTENTION  
DU GRADE DE MAÎTRE EN SCIENCES (M. Sc.)  
EN SCIENCES INFORMATIQUE**

**QUÉBEC, CANADA**

**© GAEL TCHAWÉ DISSI, 2024**

## RÉSUMÉ

L'industrie des technologies de l'information en pleine expansion et les entreprises, telles que les grands studios de jeux vidéo, doivent gérer des projets de développement de plus en plus complexes avec des équipes interdisciplinaires. De nos jours, ces équipes travaillent souvent à distance (au moins partiellement) en utilisant le cadre de gestion de projet *Scrum* (une approche adaptative de la famille Agile). Cependant, ce cadre n'est pas adapté au contexte des équipes virtuelles et plusieurs problèmes se posent dans son application depuis l'avènement du télétravail. Ces problèmes sont liés à la communication et à la confiance entre les membres de l'équipe qui ne sont pas bien abordés par les outils du cadre original. Celui-ci a été défini pour être utilisé par de petite équipe présente sur un même site avec une communication directe dans un environnement de proximité.

Pour apporter des pistes de solutions à ces enjeux, nous proposons dans ce mémoire un nouvel outil informatique exploitant les techniques d'analyse du langage naturel (Natural Language Processing ou NLP) pour améliorer le processus de gestion du projet en Scrum.

Nous proposons de tirer profit de l'outil de communication privilégié du télétravail, soit l'outil de clavardage (*chat*) par lequel tous les membres de l'équipe communiquent entre eux de manière journalière. Notre nouveau module pour la gestion de projet utilise les données des activités de communication de l'équipe sur la plateforme de chat pour extraire et appliquer une analyse quantitative de plusieurs métriques afin de déduire l'état de santé de la communication de l'équipe et de fournir des retours d'information au gestionnaire. Ce nouvel outil a été implémenté en Python et a été testé en utilisant des scénarios de communication synthétiques générés par Chat GPT.

## REMERCIEMENTS

Je souhaite tout d'abord offrir mes remerciements les plus sincères à monsieur Bruno Bouchard, qui, en tant que directeur de recherche, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi que pour l'inspiration, l'aide, les encouragements, ainsi la patience et le temps qu'il a bien voulu me consacrer.

Un grand et chaleureux merci à ma femme Blandine pour son Soutien et ses conseils. Également, merci aussi de m'avoir soutenu moralement tout au long de cette longue aventure. Merci pour sa présence à mes côtés.

Je tiens à remercier sincèrement les personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire. Ainsi, que tous mes proches de près ou de loin qui m'ont toujours soutenus et encouragés. Une pensée à mon beau-frère Serge Kinfack qui m'a financièrement soutenue pour l'obtention de mon visa afin que je réalise mon rêve de venue étudier dans cette prestigieuse université canadienne.

## LISTE DES ABRÉVIATIONS

| <b>Abréviations</b> | <b>Définitions</b>                                       |
|---------------------|--|
| ACM                 | Association of computing machinery                       |
| BERT                | Bidirectional Encoder Representation From transformation |
| CP                  | Chef de projet   |
| DES                 | Designer   |
| IA                  | Intelligence Artificielle                                |
| IEEE                | Institute of electrical and electronic engineer          |
| LLM                 | Large Model Language                                     |
| NER                 | Reconnaissance des entités nommées                       |
| NLP                 | Natural Language Processing                              |
| NLTK                | Natural Language toolkit                                 |
| NN                  | Neuronaux profonds                                       |
| POS                 | Part-of -speech tagging                                  |
| RCSPS               | Resource constrained project scheduling problem          |
| SME                 | Subject Matter Expert                                    |
| TI                  | Technologies de l 'information                           |
| UT                  | Université Du Texas                                      |
| VADER               | Valence Aware Dictionary and entiment                    |

## TABLE DE MATIÈRES

|  |            |
|--|------------|
| <b>RÉSUMÉ</b> .....  | <b>II</b>  |
| <b>REMERCIEMENTS</b> .....   | <b>III</b> |
| <b>LISTE DES ABRÉVIATIONS</b> .....  | <b>IV</b>  |
| <b>TABLE DE MATIÈRES</b> .....   | <b>V</b>   |
| <b>LISTE DES FIGURES</b> .....   | <b>VII</b> |
| CHAPITRE 1 INTRODUCTION .....  | 1          |
| CHAPITRE 2 REVUE DE LA LITTÉRATURE .....   | 5          |
| 2.1 INTRODUCTION .....   | 5          |
| 2.2 TRAVAUX SUR L'APPLICATION DU NLP AVEC DES ÉQUIPES .....                                    | 6          |
| 2.2.1 Travaux sur l'application du NLP en gestion de projet .....                              | 8          |
| 2.2.2 Travaux sur l'application du NLP avec Scrum .....  | 10         |
| 2.2.3 Conclusion .....   | 16         |
| CHAPITRE 3 UN NOUVEL OUTIL LOGICIEL BASÉ SUR LE NLP POUR AIDER<br>À LA GESTION DE PROJET ..... | 17         |
| 3.1 INTRODUCTION .....   | 17         |
| 3.2 PRÉSENTATION DE L'OUTIL ET DES FONDEMENTS<br>ALGORITHMIQUES .....                          | 19         |
| 3.2.1 Approche théorique .....   | 20         |
| 3.2.2 Implémentation de l'outil .....  | 21         |
| 3.2.3 Schéma de l'architecture de l'outil .....  | 21         |
| 3.3 EXPÉRIMENTATION ET RÉSULTATS .....   | 29         |
| 3.3.1 Protocole expérimental .....   | 30         |
| 3.3.2 Configuration de l'expérimentation .....   | 31         |
| 3.3.3 Protocole expérimental .....   | 31         |
| 3.3.4 Résultats obtenus .....  | 32         |
| 3.3.5 Phase de planification .....   | 33         |
| 3.3.6 Phase de développement .....   | 34         |
| 3.3.7 Phase de développement et correction des bugs .....                                      | 36         |
| 3.3.8 Phases de test et de livraison .....   | 37         |
| 3.4 VALEUR AJOUTÉE DE L'OUTIL .....  | 39         |
| 3.5 CONCLUSION .....   | 39         |
| CHAPITRE 4 CONCLUSION ET PERSPECTIVE .....   | 41         |
| <b>BIBLIOGRAPHIE</b> .....   | <b>43</b>  |
| <b>APPENDICE : EXEMPLES DE SCÉNARIOS DU PROJET</b> .....                                       | <b>47</b>  |

## LISTE DES FIGURES

|   |    |
|---|----|
| Figure 1 Interface utilisateur.....   | 20 |
| Figure 2 Architecture logicielle de l'outil proposé .....                   | 22 |
| Figure 3 Architecture de l'environnement de simulation pour les tests ..... | 31 |
| Figure 4 Résultats de l'analyse de la phase de planification .....          | 34 |
| Figure 5 Phases de développement.....                                       | 35 |
| Figure 6 Phases de développement et correction des bugs.....                | 37 |
| Figure 7 Phase de test et livraison .....                                   | 38 |

## CHAPITRE 1 INTRODUCTION

L'industrie des technologies de l'information (TI) est en grande expansion depuis les deux dernières décennies. Au Canada seulement, cette industrie a généré 242 milliards de dollars de revenu en 2021, avec une croissance nominale de 15,3 % de 2016 à 2021 [1]. Cependant, l'époque où les gens développaient des logiciels dans de petits bureaux est désormais révolue. De nos jours, la flexibilité et le travail à distance sont le nouveau paradigme en vogue dans l'industrie. À ce titre, on note que 67 % des employés des entreprises en TI font du télétravail en 2024 [2]. Ce mode de production est donc maintenant bien établi dans ce secteur. Le domaine se distingue également comme celui dans lequel les entreprises tentent de gérer leurs projets de développement en utilisant des processus adaptatifs, tels que le cadre Agile/Scrum [3]. Ce cadre permet de répondre à des problèmes complexes et adaptatifs tout en livrant efficacement des produits de la plus grande valeur possible [4]. Le cadre Scrum définit l'équipe Scrum. Celle-ci est autoorganisée, composée de plusieurs membres (moins de 10), chacun jouant un rôle bien défini (Scrum Master, Product Owner, ou Développeur), participant à un ensemble d'événements rituels de synchronisation régulés, et utilisant une gamme d'artefacts et de règles associées. Chaque composant de ce cadre a un objectif spécifique et est essentiel au succès et à l'utilisation de Scrum dans la gestion d'un projet.

De nombreuses études [4-7] ont été réalisées sur les problèmes fondamentaux affectant l'efficacité d'une équipe dans les projets en TI. Le manque de communication entre les membres de l'équipe, une mauvaise communication et la méfiance au sein de l'équipe semblent être parmi les principaux problèmes [7]. Ces problèmes sont amplifiés lorsque l'on ajoute le contexte d'une équipe virtuelle avec des personnes travaillant à distance.

Traditionnellement, les équipes Scrum prospèrent lorsque les membres sont colocalisés, avec des groupes soudés travaillant tous au même endroit. La colocalisation permet des contacts fréquents en personne, construit rapidement la confiance, simplifie la résolution de problèmes, encourage la communication instantanée et permet une prise de

décision rapide [8]. Par conséquent, la question demeure : comment aider à la gestion des membres de l'équipe opérant dans le cadre Scrum à distance [5] ? Plus précisément, comment la technologie et les outils automatisés peuvent supporter les membres de l'équipe et les gestionnaires dans une meilleure collaboration ?

Une avenue de réponse potentielle à cette problématique réside dans les nouveaux outils tirés de l'intelligence artificielle, tels que les outils d'analyse automatisé du langage naturel [8]. En effet, les problèmes découlant du travail en distantiel étant principalement de l'ordre de la communication, l'exploitation des outils informatisés d'analyse de la communication humaine semble une voie naturelle d'exploration. De plus formellement, les outils d'analyse du langage naturel (NLP, pour Natural Language Processing) en intelligence artificielle sont des technologies qui permettent aux machines de comprendre, interpréter et générer du langage humain. Ils sont au cœur de nombreuses applications d'IA, telles que les chatbots, la traduction automatique, l'extraction d'information, et bien d'autres. Ces outils permettent d'abord de traiter le texte brut pour le préparer à une analyse plus approfondie. Cela comprend plusieurs étapes [9] :

- **Tokenization** : Découpage du texte en unités plus petites, telles que des mots ou des phrases.
- **Lemmatisation et Stemming** : Réduction des mots à leur forme racine (par exemple, « courir » au lieu de « courait » ou « courirais »).
- **Suppression des Stop Words** : Élimination des mots courants sans signification spécifique pour l'analyse, comme « le », « la », « de », etc.

Ensuite, ces outils permettent une analyse syntaxique approfondie :

- **Part-of-Speech Tagging (POS)** : Identification des catégories grammaticales des mots (verbes, noms, adjectifs, etc.).
- **Parsing syntaxique** : Analyse de la structure des phrases pour comprendre comment les mots sont liés les uns aux autres (par exemple, sujet-verbe-objet).
- **Chunking** : Regroupement des mots en unités significatives, comme des syntagmes nominaux ou verbaux.

Enfin, ils permettent également l'analyse sémantique du contenu (la compréhension et l'interprétation des propos) :

- **Reconnaissance des entités nommées (NER)** : Identification des entités spécifiques dans le texte, telles que les noms de personnes, de lieux ou d'organisations.
- **Désambiguïsation des sens des mots** : Détermination du sens correct d'un mot qui a plusieurs significations en fonction du contexte.
- **Analyse des relations sémantiques** : Identification des relations entre les entités, comme qui fait quoi à qui.

Dans la littérature scientifique, plusieurs initiatives ont été proposées pour exploiter le NLP dans un contexte de gestion de projet [10-14]. Par exemple, Balasubramani et al. [10] ont utilisé des outils de technologie NLP pour aider à résoudre des problèmes dans la gestion de projets de construction, tels que les volumes importants de données de projet et les processus manuels sujets aux erreurs. De leur côté, Giuda et al. [11] ont tenté d'exploiter ces mêmes techniques pour la modélisation de l'information dans la gestion de grands projets, notamment pour traduire les données d'information textuel en terme numérique. Ziora [12] a fait une revue de l'utilisation des techniques NLP pour optimiser les processus d'affaires en entreprise. Du côté des approches adaptatives et du Scrum, Raharjana et al. [13] ont récemment fait une étude de l'utilisation du NLP pour améliorer la qualité des histoires utilisateurs, un artefact Scrum de planification qui s'insère dans un backlog de produit (un découpage macro des étapes de production). Enfin, ce n'est que récemment que certains chercheurs, tel que Kaleemunnisa et al. [14], ont débuté à intégrer des éléments de NLP dans l'optique de dénouer des impasses et des bloquants dans le travail des équipes Scrum.

À la lumière de nos investigations, il est devenu clair que peu de travaux avait été réalisés pour expérimenter l'exploitation des techniques NLP dans le contexte spécifique de la gestion de projets Scrum. En particulier, la majorité des travaux existants ne tentent pas de fournir de nouveaux outils d'aide au gestionnaire à partir des données de clavardage de l'équipe. Cette source d'information en télétravail est pourtant cruciale et porteuse de tant

d'éléments intéressant concernant la relation des membres de l'équipe, leur motivation, leur santé communicationnelle, leurs griefs les uns envers les autres, etc. Notre objectif dans ce travail de recherche est donc de proposer un outil d'analyse automatique de langage (NLP) qui permettra aux gestionnaires de projet d'utiliser les interactions par voie de messagerie instantané pour analyser l'état de santé communicationnelle et d'avancement du projet afin de lui permettre une meilleure prise de décision. Il vise à lui fournir un outil graphique, pouvant par exemple est pairé avec un logiciel comme Microsoft Teams.

La suite de ce mémoire est organisée de la façon suivante.

- Le chapitre 2 présente un état de l'art ciblé sur l'exploitation des techniques de NLP dans un contexte de gestion de projet.
- Le chapitre 3 décrira la contribution du mémoire et l'outil de proposé pour aider les gestionnaires en contexte de développement Scrum.
- Le chapitre 4 présente l'expérimentation et les résultats montrant le potentiel de notre approche. Cette expérimentation a été réalisé en utilisant des scénarios réalistes de conversation générés par Chat GPT.

Enfin, le dernier chapitre présente la conclusion des travaux effectués et une ouverture sur les perspectives de travaux futurs.

## **CHAPITRE 2**

### **REVUE DE LA LITTÉRATURE**

#### **2.1 INTRODUCTION**

Dans un monde où les équipes de projet jouent un rôle crucial dans l'atteinte des objectifs organisationnels et l'innovation, il est indispensable de comprendre et d'améliorer les dynamiques de communication au sein de ces équipes [15]. Une communication efficace et une collaboration harmonieuse sont des éléments clés de la performance collective. Cependant, dans un environnement de travail de plus en plus marqué par la dispersion géographique et l'utilisation prédominante de plateformes numériques pour les échanges, maintenir une communication fluide et productive constitue un défi majeur [16]. C'est particulièrement le cas dans une production exploitant le cadre de gestion de projet Scrum [17]. C'est dans ce contexte que le traitement automatique du langage naturel (NLP) s'impose comme une technologie prometteuse pour analyser et optimiser les interactions entre les membres d'une équipe de projet.

L'analyse automatique du langage, une sous-discipline de l'intelligence artificielle, se concentre sur l'interaction entre les ordinateurs et le langage humain [46]. Son objectif est de permettre aux machines de comprendre, interpréter et répondre au langage naturel de manière fluide. Les avancées récentes en NLP ont engendré des développements significatifs dans divers domaines, tels que la traduction automatique, l'analyse des sentiments, les assistants vocaux, et la reconnaissance vocale [8]. Ces progrès offrent des opportunités inédites pour améliorer la gestion des équipes de projet, en permettant une analyse plus précise et rapide des communications internes [3].

Reconnaître les interactions entre les membres d'une équipe est crucial pour plusieurs raisons [33]. Des recherches ont démontré que les équipes bénéficiant d'une communication efficace et d'une bonne dynamique de collaboration sont non seulement plus productives, mais aussi plus innovantes [15]. Edmondson [18] a mis en avant l'importance de la sécurité psychologique et de la communication ouverte au sein des équipes pour créer un environnement propice à l'apprentissage et à une performance accrue. Cependant, dans des

environnements de travail complexes et diversifiés, maintenir un tel niveau de communication et de collaboration peut s'avérer difficile. Les outils traditionnels de gestion de projet échouent souvent à capturer les subtilités des interactions humaines, ce qui peut conduire à des malentendus et des conflits [16].

Dans ce contexte, l'analyse automatique du langage se révèle être une solution intéressante. Les techniques de NLP, telles que l'analyse des sentiments, la reconnaissance des entités nommées et l'analyse des réseaux de communication, permettent d'extraire des informations essentielles des échanges textuels et vocaux entre les membres de l'équipe [40]. Par exemple, les modèles de langage avancés comme BERT [19] et les architectures de transformateurs [20] offrent une compréhension approfondie et contextuelle des communications humaines. Ces modèles sont capables d'identifier les sentiments, de détecter les interactions conflictuelles et même de prédire les dynamiques de collaboration au sein de l'équipe.

## **2.2 TRAVAUX SUR L'APPLICATION DU NLP AVEC DES ÉQUIPES**

Les applications du traitement automatique du langage (NLP) dans la reconnaissance des interactions au sein des équipes sont diverses et en constante évolution [43].

Par exemple, S. Katada et al. [21] ont exploré l'utilisation de la NLP pour analyser les discussions en ligne, les courriels et les enregistrements de réunions afin d'identifier les sentiments et les émotions exprimés par les membres de l'équipe. Ces analyses permettent de détecter les tensions et les conflits potentiels, offrant ainsi aux gestionnaires de projet la possibilité d'intervenir de manière proactive. Plus spécifiquement, les chercheurs ont examiné les effets des signaux physiologiques dans l'analyse multimodale des sentiments, à partir du texte écrit, en évaluant tous les modèles de fusion pour différents types d'estimation des sentiments dans des contextes d'interaction humain-agent naturalistes. Les résultats qu'ils ont obtenus suggèrent que les caractéristiques physiologiques sont efficaces dans le modèle unimodal et que la fusion des représentations linguistiques avec les caractéristiques physiologiques fournit les meilleurs résultats pour estimer les étiquettes de sentiments personnels telles qu'annotées par les utilisateurs eux-mêmes. En revanche, la fusion

tensorielle des représentations linguistiques avec les caractéristiques audiovisuelles est efficace pour estimer les étiquettes de sentiments telles qu'annotées par un tiers dans les tâches de régression, ce qui peut être dérivé des signaux correspondants observables par les humains. Une analyse détaillée des résultats d'estimation des sentiments personnels suggère que différentes modalités jouent des rôles distincts dans l'estimation des sentiments, et les implications correspondantes sont discutées. Au final, cette étude est intéressante pour montrer le potentiel d'exploitation du NLP pour mettre en lumière les sentiments des membres de l'équipe à partir des communications écrites. Cependant, le contexte de cette étude diffère du nôtre. Les communications utilisées étaient principalement asynchrones. Par ailleurs, l'étude ne s'est pas faite dans un contexte de gestion de projet Scrum avec des éléments de télétravail. Enfin, bien que l'interprétation des sentiments des membres soit intéressante, elle ne permet pas d'offrir, sans représentation visuelle synthétique, un outil utile et efficace dans un contexte de gestion.

D'un autre côté, V. Yadav et S. Bethard [7] présente dans leurs travaux une revue complète des architectures de réseaux neuronaux profonds pour la reconnaissance des entités nommées (NER). La reconnaissance des entités nommées (NER) est un composant clé des systèmes de traitement du langage naturel pour la réponse aux questions, la recherche d'information, l'extraction de relations, etc. Les systèmes NER ont été largement étudiés et développés depuis des décennies, mais les systèmes précis utilisant des réseaux neuronaux profonds (NN) n'ont été introduits que ces dernières années. Dans leur étude, les chercheurs ont comparé avec les approches antérieures basées sur l'ingénierie des caractéristiques et d'autres algorithmes d'apprentissage supervisé ou semi-supervisé. Leurs résultats soulignent les améliorations réalisées par les réseaux neuronaux et montrent comment l'incorporation de certaines leçons tirées des travaux précédents sur les systèmes NER basés sur les caractéristiques peut conduire à des améliorations supplémentaires. Finalement, ils ont montré que la reconnaissance des entités nommées peut aider à clarifier les rôles et les responsabilités des membres de l'équipe, facilitant ainsi une meilleure répartition des tâches et une compréhension plus précise des contributions de chacun. Cependant, ces travaux ne se sont pas déployés dans un contexte Agile, et ils n'exploitent pas les données de clavardage.

Hovy et Spruit [23] ont quant à eux proposé des travaux sur l'implication éthique de l'utilisation de la NLP pour l'analyse des données textuelles, notamment en ce qui concerne la confidentialité et les questions éthiques liées à la surveillance. Le traitement du langage naturel (NLP) impliquait autrefois principalement des corpus anonymes, dans le but d'enrichir l'analyse linguistique, et était donc peu susceptible de soulever des préoccupations éthiques. Cependant, avec la généralisation du NLP et l'utilisation croissante de données provenant des médias sociaux, la situation a changé : les résultats des expériences et des applications NLP peuvent désormais avoir un impact direct sur la vie des utilisateurs individuels. Jusqu'à présent, le discours sur ce sujet dans le domaine n'a pas suivi le développement technologique, tandis que le discours public était souvent centré sur des dangers exagérés. Dans leurs travaux, les chercheurs soulèvent un certain nombre d'implications sociales du NLP et discutent de leur signification éthique, ainsi que des moyens de les aborder. Ce travail est corolaire au nôtre, car même s'il n'aborde pas directement la problématique soulevée, il indique des pistes pour une utilisation responsable des approches NLP dans des contextes réels pouvant impliquer des données personnelles, comme des conversations privées.

Cependant, malgré ces avancées prometteuses, l'application de la NLP dans ce domaine présente également des défis importants. La confidentialité des données est une préoccupation majeure, car les communications internes des équipes contiennent souvent des informations sensibles. De plus, les modèles de NLP doivent être capables de saisir les nuances culturelles et linguistiques, ce qui peut être particulièrement complexe dans des équipes internationales et diversifiées.

### **2.2.1 Travaux sur l'application du NLP en gestion de projet**

De manière plus spécifique, plusieurs travaux ont été faits sur l'exploitation de l'analyse du langage pour augmenter l'efficacité dans un contexte de gestion de projet. Cette section présente ces travaux.

Les récents travaux de Shai et al. [23] sont particulièrement intéressants. Dans cet article, ils explorent le potentiel de l'utilisation du traitement du langage naturel (NLP) pour

résoudre l'un des problèmes cruciaux en gestion de projet : les problèmes de planification de projets avec contraintes de ressources (RCPSP). Les méthodes conventionnelles utilisées pour résoudre les RCPSP reposent sur des modèles mathématiques nécessitant des entrées de données structurées, ce qui peut être chronophage et sujet à des erreurs. Les auteurs de l'article cherchent à démontrer les avantages de l'utilisation du NLP pour extraire et traiter les données de projet à partir de sources non structurées, telles que les plans de projet rédigés en langage naturel, afin d'améliorer la précision et l'efficacité des solutions de planification. L'étude conclut que le NLP peut jouer un rôle significatif dans la résolution des RCPSP et ouvre de nouvelles perspectives pour la recherche future dans ce domaine. L'article contribue à la littérature en soulignant l'importance du NLP dans la planification de projets et son potentiel à fournir des solutions plus efficaces pour les projets soumis à des contraintes de ressources. L'esprit de cet article va dans le sens de nos travaux, c'est-à-dire qu'il cherche à trouver des façons d'améliorer l'efficacité de la gestion de projet en exploitant le NLP. Cependant, ces travaux couvrent un autre aspect de la gestion de projets, soit la planification et l'échéancier, alors que nos travaux s'attardent plus précisément aux problèmes de motivation, de communication et de coordination de l'équipe. Par ailleurs, ces travaux n'ont pas été réalisés dans un contexte Scrum avec des éléments de télétravail.

Le domaine de la construction est riche en projets d'envergure, des projets complexes avec de grands chantiers. Il n'est donc pas surprenant de voir plusieurs de travaux [10-12, 25] portent sur ce secteur. Dans leur récente étude de 2024, Hussain et al. [25] font une revue de la littérature portant sur l'automatisation dans la construction pour améliorer l'efficacité de ce secteur. Ils relèvent de nombreux progrès qui ont été réalisés dans ce domaine, et l'utilisation du traitement du langage naturel (NLP) a suscité un intérêt croissant ces dernières années. Cette revue systématique de la littérature examine les contributions faites au cours de la dernière décennie dans le domaine spécifique de la mise en œuvre des modèles NLP dans l'industrie de la construction commerciale pour aider à améliorer l'efficacité en gestion de projet. La revue se concentre sur l'identification des efforts de recherche pertinents et de leurs contributions, l'analyse des techniques NLP utilisées et des résultats obtenus, ainsi que sur la détermination des lacunes et des limitations basées sur la littérature. Les articles

examinés ont été sélectionnés parmi un total de 565 enregistrements récupérés dans les bases de données Scopus, ASCE et Porques. Les articles ont ensuite été filtrés, classifiés, résumés et analysés. La revue a permis d'identifier les articles les plus intéressants dans lesquels les technologies développées ont le potentiel d'applications immédiates dans différentes phases de projet — de la documentation et des appels d'offres, à la clôture du projet, jusqu'à la gestion des installations post-construction. Il a été conclu que l'amélioration et l'automatisation des processus dans ces phases de projet à l'aide du NLP conduisaient à une augmentation de l'efficacité du projet. Certaines lacunes et limitations dans la revue de la littérature ont été identifiées, et des recommandations pour des recherches futures ont été fournies. On note, cependant, que bien que l'ensemble de la chaîne de projet ait été traitée dans les travaux, de l'analyse des besoins du client jusqu'à la clôture, aucun des travaux cités dans l'étude ne portait sur l'analyse des interactions et des communications entre les membres des équipes projet. Par conséquent, on note que cet aspect correspond souvent à un angle mort dans bon nombre de travaux.

D'autres travaux, comme ceux de Kang et al. [26], ont passé en revue les articles de la Liste UT Dallas des 24 principales revues de commerce qui utilisent le NLP comme technique analytique principale pour illustrer comment les données textuelles peuvent être exploitées pour faire progresser les théories de gestion dans plusieurs disciplines. Ils décrivent les ensembles d'outils disponibles et les étapes procédurales pour utiliser le NLP comme technique analytique, ainsi que ses avantages et ses inconvénients. Ce faisant, ils mettent en évidence les défis managériaux et technologiques associés à l'application du NLP dans la recherche en gestion afin de guider les futures investigations. En revanche, cette étude et la précédente s'insèrent dans un contexte d'approche de gestion de projet dite « prédictive » qui, à l'inverse des approches Agiles, dites « adaptatives », comporte moins d'interactions soutenues sur le plan communicationnel.

### **2.2.2 Travaux sur l'application du NLP avec Scrum**

Le Scrum [6] est l'approche de type Agile [27] la plus utilisée dans l'industrie du numérique [2]. De manière plus spécifique, les approches de type Agile s'appuient sur les

préceptes du manifeste Agile [27] et proposent des lignes directrices de gestion de projet qui se distinguent par la flexibilité et l'adaptabilité face aux changements. Contrairement aux méthodes traditionnelles, telles que le modèle en cascade [28] ou les approches prédictives [29], l'Agile se concentre sur des cycles de développement itératifs et incrémentaux, permettant une révision constante et une adaptation aux besoins changeants des clients. Le concept repose sur les principes suivants :

**Collaboration avec le client** : Plutôt que de se concentrer uniquement sur la documentation, l'approche Agile favorise une collaboration continue avec le client pour s'assurer que le produit final répond à ses attentes et besoins.

**Adaptabilité aux changements** : Les exigences peuvent évoluer au cours du projet. L'Agile permet d'ajuster les priorités et les tâches en fonction des retours et des changements de contexte.

**Livraisons fréquentes** : Les projets sont divisés en petits morceaux appelés « itérations » ou « sprints », avec des livraisons fréquentes de produits ou de fonctionnalités, permettant de recueillir des retours précoces et de corriger rapidement les erreurs.

**Travail en équipe autoorganisée** : Les équipes sont encouragées à organiser leur propre travail et à collaborer efficacement pour atteindre les objectifs du projet.

Le Scrum [4], quant à lui, est un cadre de gestion de projet de type Agile spécifiquement conçu pour faciliter le développement itératif et incrémental. Il fournit une structure pour organiser et gérer le travail en équipe, en mettant l'accent sur la livraison de valeur à chaque itération. Le Scrum organise le travail en petites équipes de moins de 10 personnes dans lesquelles on retrouve trois rôles clés :

**Scrum Master** : Responsable de l'application du cadre Scrum et de la suppression des obstacles qui entravent le progrès de l'équipe.

**Product Owner** : Représente les parties prenantes et est responsable de la définition et de la priorisation des exigences du produit dans le *backlog*.

**Équipe de Développement** : Composée de professionnels qui travaillent ensemble pour livrer les fonctionnalités du produit à chaque sprint.

Le Scrum propose l'utilisation d'artefacts permettant la coordination de l'équipe et la planification du projet. Les artefacts sont les suivants :

**Product Backlog** : Une liste ordonnée de toutes les fonctionnalités, améliorations et corrections à apporter au produit. C'est un document évolutif qui reflète les priorités du produit. Les fonctionnalités sont décrites sous forme d'histoires utilisateurs. Ces histoires décrivent une fonctionnalité utile pour l'utilisateur qui peut être testée à la fin d'un Sprint. Il s'agit du découpage du projet en « gros » morceaux ou de la planification macro du projet.

**Sprint Backlog** : Une liste des tâches et des objectifs à atteindre pendant une itération (un Sprint de 2 à 4 semaines). C'est le découpage en petites tâches utilisé pour répartir le travail durant un sprint. C'est la planification micro du morceau de projet à produire maintenant.

**Increment** : La somme de toutes les fonctionnalités complètes et intégrées qui sont prêtes à être livrées à la fin d'un sprint.

Enfin, le Scrum utilise des rituels ou événements permettant de cadencer, de réguler et de coordonner les actions de l'équipe projet. Ces rituels sont les suivants :

**Sprint** : Une période de développement fixe (typiquement de 2 à 4 semaines) au cours de laquelle un incrément de produit est créé.

**Sprint Planning** : Une réunion au début de chaque sprint pour définir les objectifs du sprint et planifier les tâches nécessaires. C'est à ce moment que le découpage (sprint backlog) est créé en équipe.

**Daily Scrum** : Une courte réunion quotidienne (15 minutes) où l'équipe discute des progrès, des obstacles rencontrés et des plans pour la journée.

**Sprint Review** : Une réunion à la fin du sprint où l'équipe présente les résultats du sprint aux parties prenantes pour obtenir des retours. L'objectif de cette réunion est de s'assurer de la valeur créée dans l'incrément de produit.

**Sprint Retrospective** : Une réunion de réflexion après chaque sprint pour identifier les points forts, les points à améliorer et ajuster les pratiques de travail de l'équipe.

Scrum fournit une structure et des processus pour mettre en œuvre les principes Agile de manière efficace. En utilisant Scrum, les équipes peuvent gérer les projets de manière flexible, améliorer continuellement leur processus de développement et garantir une livraison régulière de valeur aux clients.

Bien que la littérature concernant l'exploitation du NLP dans un contexte spécifique de gestion de projet en Scrum soit limitée, certains travaux [14] ont abordé la question et proposé des solutions innovantes pour améliorer la production en Scrum. L'un des travaux récents que l'on peut citer est celui de Kaleemunnisa et al. [14]. Dans leurs recherches, ils se sont concentrés sur les obstacles rencontrés par les étudiants dans les projets de fin d'études utilisant la méthodologie Scrum. Des notes de réunions Scrum ont été collectées dans un ensemble de données pour permettre aux rôles Scrum et aux instructeurs de suivre les progrès et les problèmes. Ils ont ainsi pu identifier 9 catégories d'obstacles dans cet ensemble de données : Android, Compétences en codage, Débogage, Facteurs externes, Firebase/Base de données, Git/GitHub.

Travail en équipe, Gestion du temps, et Conception UI/UX. Ils ont ensuite proposé l'utilisation d'un Modèle de Langage Large (LLM) pour classifier ces obstacles. Les résultats ont permis de montrer que le traitement du langage naturel (NLP) a le potentiel de soutenir les processus d'ingénierie logicielle dans un contexte Scrum. La nouveauté de cette recherche réside dans le fait qu'elle tente d'identifier les obstacles rencontrés par les équipes Scrum d'étudiants à l'aide de l'IA et de soutenir les étudiants et les instructeurs, ce qui va dans le même sens que notre proposition. La pertinence de l'approche a été discutée avec des experts en la matière (SME) de l'industrie. Le modèle proposé est utile tant dans les environnements

académiques qu'industriels pour identifier en temps réel les domaines nécessitant une attention et, s'ils sont corrigés, pour augmenter la productivité de l'équipe. Cependant, comme on peut le constater, l'outil proposé porte uniquement sur les problèmes « techniques » liés au développement, tels que les bogues dans le codage ou la mauvaise configuration d'outils informatiques. L'aspect sur l'interaction entre les membres de l'équipe n'est pas du tout abordé, ni l'aspect portant sur le télétravail.

De leur côté, Raharjana et al. [13] ont tenté de recenser et d'analyser l'ensemble des travaux ayant exploité le NLP pour améliorer la création des histoires d'utilisateurs dans un contexte de production Scrum. Leur objectif spécifique était de réaliser une revue systématique de la littérature pour capturer l'état actuel de la recherche en NLP sur les histoires d'utilisateurs. La stratégie de recherche utilisée pour obtenir des articles pertinents a été de consulter les références SCOPUS, ScienceDirect, IEEE Xplore, ACM Digital Library, SpringerLink et Google Scholar. Des critères d'inclusion et d'exclusion ont été appliqués pour filtrer les résultats de la recherche. Les auteurs ont également utilisé les techniques de snowballing avant et arrière pour obtenir des résultats plus complets. Les résultats de la recherche ont identifié 718 articles publiés entre janvier 2009 et décembre 2020. On note que, malgré la somme des articles, il s'agit d'un faible échantillon pour une si longue période, montrant que le sujet est peu abordé. Après application des critères d'inclusion/exclusion et de la technique de snowballing, ils ont finalement identifié 38 études principales qui discutent des techniques NLP dans les histoires d'utilisateurs. Encore une fois, sur 10 ans, 38 articles recensés correspondent à un secteur peu étudié. La plupart des études ont utilisé les techniques NLP pour extraire des aspects de qui, quoi et pourquoi à partir des histoires d'utilisateurs. Les objectifs de ces différents travaux sur le NLP dans les histoires d'utilisateurs étaient variés, allant de la découverte de défauts, à la génération d'artefacts logiciels, en passant par l'identification de l'abstraction clé des histoires d'utilisateurs et la traçabilité des liens entre les modèles et les histoires d'utilisateurs. Au final, ce travail important a permis de montrer que le NLP peut aider les analystes système à gérer les histoires d'utilisateurs. La mise en œuvre du NLP dans les histoires d'utilisateurs présente de

nombreuses opportunités et défis. Il est nécessaire d'explorer les techniques NLP et d'utiliser des méthodes d'évaluation rigoureuses pour obtenir des recherches de qualité.

Enfin, on peut terminer cette section en notant les récents travaux tout juste publiés en juillet 2024 de Ademola Fathia [30]. Dans ses travaux, elle propose l'intégration de l'intelligence artificielle (IA) dans les méthodologies de développement Agile pour offrir une opportunité significative d'améliorer l'efficacité, la précision et l'adaptabilité des processus de développement logiciel. Cette recherche explore l'application au sens large de l'IA, mais aussi du NLP, dans les deux principaux cadres Agile : Scrum et Kanban. Les méthodologies Agile comportent des défis tels que la gestion des charges de travail, la prévision des délais de projet et l'optimisation de l'allocation des ressources persistent. Les technologies IA, avec leurs capacités en analyse de données, reconnaissance de motifs et modélisation prédictive, offrent des solutions prometteuses à ces défis. Dans Kanban, avec son accent sur la livraison continue et le flux, il est proposé de bénéficier de l'IA pour optimiser la gestion des flux de travail. Les algorithmes d'apprentissage automatique peuvent ainsi prédire les temps de réalisation des éléments de travail, identifier les retards potentiels et recommander des actions pour maintenir un flux constant de tâches. L'IA est aussi proposé pour améliorer les outils de gestion visuelle en ajustant dynamiquement les tableaux Kanban en fonction des données en temps réel, garantissant ainsi que les membres de l'équipe disposent d'informations à jour pour prendre des décisions éclairées.

Dans Scrum, l'auteur propose que l'IA soit exploitée pour automatiser la planification des sprints, la priorisation des backlogs et l'estimation des tâches, réduisant ainsi les biais humains et améliorant la précision. Les analyses basées sur l'IA peuvent fournir des indications sur la performance de l'équipe, identifier les goulets d'étranglement et suggérer des ajustements pour améliorer la productivité. De plus, il est proposé d'utiliser le traitement du langage naturel (NLP) pour faciliter une communication plus efficace au sein des équipes Scrum en analysant et en résumant les réunions quotidiennes, les revues de sprint et les rétrospectives. En ce sens, il s'agit du travail de recherche recensé qui s'approche le plus de notre proposition. Cependant, elle ne s'est pas attaquée à l'exploitation des données en provenance des séances de clavardage en provenance des membres de l'équipe. Pourtant,

comme mentionné en introduction, cet espace de communication contient un ensemble de données précieuses qui, s'y elles sont bien exploitées, peuvent nous communiquer une information riche concernant la communication, la motivation et les relations entre les membres de notre équipe de travail.

### **2.2.3 Conclusion**

En résumé, nous avons fait une revue synthétique des travaux portant sur l'exploitation du NLP dans différents contextes liés d'une façon ou d'une autre à notre problématique de gestion de projet en Scrum. Bien que la littérature soit limitée, plusieurs travaux ont montré l'intérêt d'exploiter les techniques de NLP pour améliorer l'efficacité de la production et de la gestion des projets, que ce soit avec des approches prédictives ou Agiles [44-46]. Cette revue de la littérature a permis de constater que les travaux que nous proposons s'inscrivent dans l'originalité, puisque peu de travaux se sont attardés à la problématique de l'utilisation du NLP avec un cadre Scrum, mais qu'en plus, aucun d'entre eux n'a tenté d'exploiter les données en provenance du clavardage d'équipe pour fournir des outils aux gestionnaires quant à la santé communicationnelle et à la motivation de l'équipe. Ainsi, cet aspect spécifique fait partie de la contribution du mémoire.

Dans les prochaines sections, nous présenterons notre proposition, les techniques exploitées, l'implémentation de l'outil, ainsi que les résultats d'une phase d'expérimentation avec des données de conversation simulées avec Chat GPT.

## CHAPITRE 3

### UN NOUVEL OUTIL LOGICIEL BASÉ SUR LE NLP POUR AIDER À LA GESTION DE PROJET

#### 3.1 INTRODUCTION

Dans ce chapitre, nous présentons notre proposition d'outil logiciel basé sur les approches d'analyse automatique du langage (NLP) qui permet d'aider à la gestion de projet en Scrum. Nous présenterons plus spécifiquement l'architecture logicielle de celui-ci, l'implémentation, le contexte expérimental et les résultats d'une phase de validation. Cet outil a pour objectif d'aider les gestionnaires de projet à mieux comprendre et évaluer les dynamiques communicationnelles au sein des équipes, en se basant sur les échanges effectués dans les canaux de discussion en ligne, comme les chats. Il vise à identifier les signaux de cohésion, de conflit et de santé communicationnelle dans les interactions, permettant ainsi une meilleure prise de décision. Pour la validation de cet outil, nous avons choisi d'utiliser **Chat GPT** [31] pour simuler des scénarios (4) de conversation entre différents membres d'une équipe de projet. Ces scénarios, qui reproduisent les échanges d'une équipe Scrum travaillant sur le développement d'un jeu vidéo, ont été soigneusement conçus pour refléter des situations réalistes rencontrées lors de projets de développement logiciel. Cette approche nous a permis de tester et valider les fonctionnalités de notre outil dans un environnement contrôlé, en générant un large éventail d'interactions, allant des échanges harmonieux aux conflits ouverts. Les scénarios simulés ont été élaborés pour représenter diverses phases clés du développement d'un projet de jeu vidéo, en s'inspirant des méthodologies agiles, et plus particulièrement du cadre Scrum. Chaque phase du projet, de la planification initiale à la livraison finale, est reproduite à travers des dialogues entre les membres d'une équipe projet typique. Ces membres incluent un **chef de projet (CP – Scrum Master)**, chargé de la coordination globale, un **développeur (DEV)** responsable de l'implémentation technique, un **designer (DES)** focalisé sur l'aspect créatif et visuel, et un **testeur (TEST)** en charge de la validation des fonctionnalités et de la qualité du produit. Les dialogues ont été structurés pour simuler les interactions quotidiennes que l'on pourrait rencontrer dans ces différentes phases : la planification des sprints, la collaboration entre les membres durant le développement, la

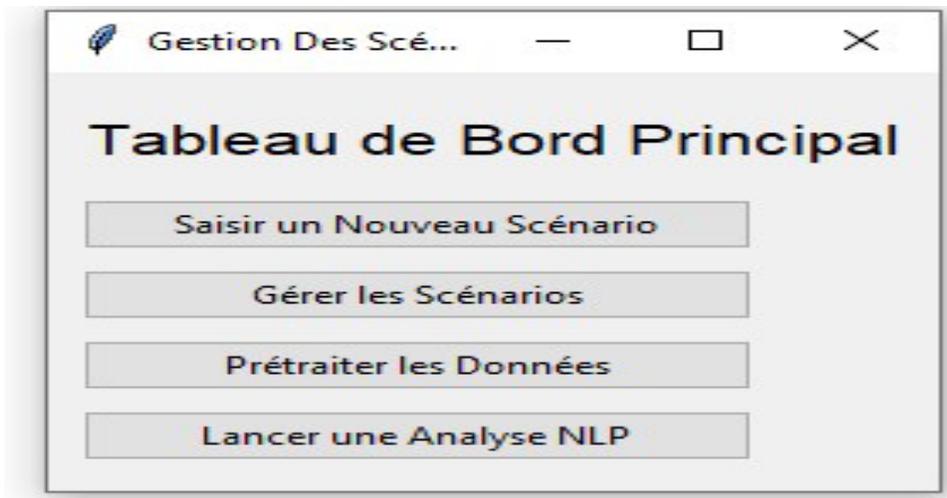
résolution de bugs identifiés pendant la phase de test, ainsi que les discussions autour de la livraison finale du produit. Ces échanges reflètent à la fois les situations positives, telles qu'une bonne cohésion d'équipe, et les situations plus complexes, impliquant des désaccords ou des conflits sur les tâches ou les délais. Cela permet de tester la capacité de l'outil à analyser et à détecter différents signaux communicationnels au sein de l'équipe. Dans le but d'évaluer la robustesse de l'outil dans des conditions diverses, nous avons inclus plusieurs situations problématiques dans les scénarios. Par exemple, dans 1 des 4 scénarios, des conflits entre membres de l'équipe ont été introduits, concernant notamment des désaccords sur la répartition des tâches et des problèmes de priorités. D'autres scénarios incluaient des situations de retard dans la livraison des tâches, causant de la frustration au sein de l'équipe. L'expérimentation s'est appuyée sur **4 scénarios distincts**, chacun représentant une phase différente du développement d'un projet de jeu vidéo. Au total, ces scénarios ont généré environ **1 200 messages**, répartis sur les différentes phases du projet. Pour chaque scénario, une équipe simulée composée de **6 à 8 membres** a été mise en place, avec des rôles variés tels que Scrum Master ou chef de projet, Designer, Développeurs, Testeurs, et autres acteurs pertinents du projet. Les types de conversations générées couvrent une large gamme de situations de gestion de projet. Parmi ces échanges, on retrouve des discussions autour de la phase de planification, développement, Développement et correction des bugs et la phase de test et livraison, ainsi que des réactions émotionnelles aux différents événements du projet. Ces scénarios ont été structurés pour refléter les étapes typiques d'un projet en méthodologie agile, telles que la planification du sprint, la réunion de revue de sprint, la phase de développement, la résolution des bugs, et la livraison finale. Ces différentes phases permettent de tester la capacité de l'outil à analyser les dynamiques de communication dans un cadre réaliste et à identifier les tensions, conflits ou succès au sein des équipes.

### 3.2 PRÉSENTATION DE L'OUTIL ET DES FONDEMENTS ALGORITHMIQUES

Dans cette section, nous détaillons l'approche algorithmique qui sous-tend l'outil que nous avons développé pour analyser les interactions d'une équipe Scrum en utilisant le traitement automatique du langage naturel (NLP). Cet outil vise à offrir aux gestionnaires de projet un aperçu approfondi des dynamiques communicationnelles au sein de leurs équipes, en analysant les échanges sur les plateformes de communication, telles que les chats ou messageries instantanées, souvent utilisées pour les discussions quotidiennes en milieu de travail.

L'algorithme exploite des techniques classiques de NLP pour traiter les messages de chat en identifiant des indicateurs clés, tels que les sentiments exprimés dans les échanges, la reconnaissance d'entités nommées (par exemple, noms de personnes, tâches, dates), et l'analyse syntaxique des phrases. En combinant ces analyses, l'outil est capable de fournir des métriques objectives sur l'état de la communication au sein de l'équipe, telles que la cohésion, les signes de conflit, ou encore la santé générale des interactions.

L'analyse des sentiments permet de déterminer si les messages sont positifs, négatifs ou neutres, offrant ainsi une vision claire de l'humeur de l'équipe. La reconnaissance d'entités nommées permet d'identifier les acteurs principaux et les tâches mentionnées dans les discussions, tandis que l'analyse syntaxique aide à comprendre les relations entre les différents éléments de la phrase, ce qui est crucial pour détecter les tensions sous-jacentes ou les ambiguïtés. L'outil a été développé en *Python* [32], utilisant des bibliothèques NLP populaires comme *spacy* et *VADER* [34]. Il intègre ces analyses dans un tableau de bord interactif où les gestionnaires peuvent suivre, en temps réel, les évolutions communicationnelles de leur équipe, identifier les problèmes émergents et ajuster leurs actions en conséquence.



*Figure 1 Interface utilisateur*

La figure 1 montre l'interface utilisateur qui permet de tester l'outil. On peut sélectionner et charger, à l'aide du premier bouton, un scénario (un scénario d'échange sur le chat), on peut gérer les différents scénarios, par exemple en ajouter ou en retirer avec le second bouton. Le prétraitement des données (nettoyage, combler les blancs, etc.) se fait à l'aide du troisième bouton. Enfin, on peut lancer une analyse NLP du scénario chargé et générer les tableaux de bord de gestion à l'aide du dernier bouton.

### **3.2.1 Approche théorique**

Ici nous avons opté pour une approche quantitative basée sur l'extraction de mots-clés spécifiques, permettant de capturer des informations sur l'état émotionnel et la qualité des échanges entre les membres de l'équipe. Pour ce faire, nous avons utilisé un dictionnaire prédéfini de 56 mots-clés représentant des émotions, des niveaux de motivation, ainsi que des indicateurs de collaboration et de blocages potentiels. Ce dictionnaire inclut des mots tels que « heureux », « bloqué », « urgent », « besoin d'aide », et d'autres termes fréquemment utilisés dans les communications de projet. Ensuite, nous avons appliqué une analyse de fréquence des termes (TF - Term Frequency) pour identifier la récurrence des mots-clés dans les discussions. Ces données sont ensuite enrichies par une analyse des sentiments basée sur un modèle préentraîné de classification des sentiments, capable de catégoriser les phrases en trois catégories : positif, neutre, ou négatif. Pour une analyse plus approfondie, nous avons également utilisé des techniques de *parsing* syntaxique pour évaluer la structure

grammaticale des phrases et déduire des indices supplémentaires sur les relations de pouvoir, les responsabilités et l'engagement des membres de l'équipe.

### 3.2.2 Implémentation de l'outil

L'outil a été implémenté en *Python* en utilisant des bibliothèques NLP populaires comme NLTK (Natural Language Toolkit) et *spaCy*. Voici les principales étapes de l'implémentation :

**Tokenization** : Nous avons utilisé la bibliothèque *spaCy* pour découper les phrases en mots (tokenization) et identifier les mots-clés. Ensuite, nous avons appliqué un traitement supplémentaire pour normaliser le texte, comme la conversion en minuscules et la suppression des mots inutiles (stop words).

**Analyse des Sentiments** : L'analyse des sentiments a été effectuée en utilisant la bibliothèque VADER (Valence Aware Dictionary and Entiment Reasoner), particulièrement adaptée pour l'analyse de textes courts et informels, comme les messages de chat. Ce modèle nous a permis de générer des scores de sentiment pour chaque message.

**Reconnaissance d'entités nommées (NER)** : La reconnaissance d'entités nommées (NER) a été réalisée à l'aide de *spaCy* pour identifier des noms de personnes, d'organisations ou de dates dans les messages. Cela permet d'analyser les responsabilités et les interactions directes entre les membres de l'équipe.

**Visualisation des Résultats** : Les résultats sont ensuite présentés dans un tableau de bord graphique, généré à l'aide de la bibliothèque *matplotlib*. Ce tableau de bord affiche des visualisations des émotions globales de l'équipe, des points de blocage détectés et des tendances de communication.

### 3.2.3 Schéma de l'architecture de l'outil

La figure 2 nous montre l'architecture générale de notre outil logiciel et présente les composants clés de l'outil. Nous décrirons chacun de ces composants dans les prochaines sections.

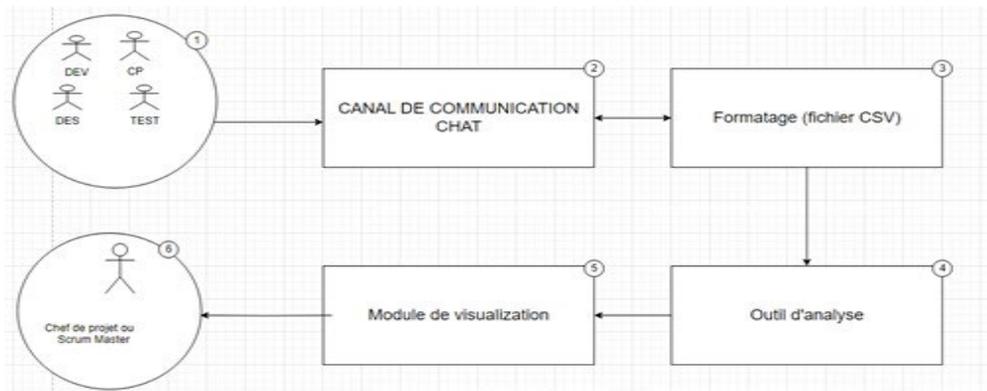


Figure 2 Architecture logicielle de l'outil proposé

## 1. Production des messages

Les données brutes des messages sont produites par les membres de l'équipe durant leurs conversations régulières et sont envoyées par les humains sur la plateforme de chat. L'équipe projet est composée de plusieurs membres ayant des rôles bien définis et complémentaires, chacun apportant une contribution essentielle au bon déroulement du projet. Dans cet exemple, l'équipe se divise en quatre groupes principaux : les développeurs (DEV), le chef de projet (CP – Scrum Master), les designers (DES) et les testeurs (TEST). Chacun de ces rôles est chargé d'une responsabilité spécifique au sein du projet, et leurs interactions sont indispensables pour garantir l'avancement harmonieux des travaux. Les développeurs (DEV) sont responsables de la programmation et de la mise en œuvre des fonctionnalités du projet. Ils créent les solutions techniques nécessaires et veillent à ce que le code réponde aux exigences fonctionnelles. Le chef de projet (CP), quant à lui, assure la coordination générale. Il supervise l'avancement global, gère les délais et les ressources, et veille à ce que le projet respecte les objectifs et le calendrier prévus. Il est souvent l'interface entre l'équipe technique et les parties prenantes extérieures. Les designers (DES) jouent un rôle clé dans la conception visuelle du projet. Ils sont chargés de créer des maquettes, des interfaces utilisateurs intuitives et esthétiques, et veillent à ce que l'apparence du produit final soit cohérente avec les objectifs du projet. Enfin, les testeurs (TEST) sont chargés de la qualité du projet. Ils s'assurent que les fonctionnalités développées fonctionnent correctement en effectuant des tests rigoureux, détectant les bugs, et garantissant que chaque élément du projet est conforme aux attentes avant la mise en production. Cette étape, centrée sur les

interactions entre les différents membres de l'équipe, est cruciale car elle constitue la source première de toutes les données nécessaires à l'analyse ultérieure. Les communications et les collaborations quotidiennes entre les développeurs, le chef de projet, les designers et les testeurs génèrent une richesse d'informations sur la dynamique de l'équipe, les obstacles rencontrés et les réussites obtenues. Chaque interaction, qu'il s'agisse d'une question posée, d'une solution trouvée ou d'une tâche accomplie, contribue à former un ensemble de données brutes qui seront ensuite analysées pour en tirer des insights. Cette première étape consiste à capturer les échanges au sein de l'équipe projet. Ces interactions sont essentielles non seulement pour l'avancement du projet, mais également pour la collecte des informations nécessaires à l'analyse de la performance globale de l'équipe et à l'évaluation de l'efficacité de la collaboration. Ces données serviront de base pour alimenter les processus d'analyse de l'outil, fournissant ainsi une vue d'ensemble des dynamiques de l'équipe et de la progression du projet.

## **2. Extraction (Données des conversations)**

Extraction des messages à partir de la plateforme de chat. Ces données sont sélectionnées et extraites de la plateforme de chat. Le canal de communication chat représente une étape cruciale dans ce processus d'aide à la décision, car il constitue la plateforme centrale où toutes les interactions entre les membres de l'équipe se déroulent. Utilisé comme principal moyen de communication pour les discussions liées au projet, cet espace de chat permet une collaboration fluide et instantanée entre les membres de l'équipe, tels que les développeurs, les testeurs, le Scrum Master, et le designer. Contrairement aux méthodes plus formelles telles que les courriels ou les réunions physiques, la communication par chat est plus directe, souvent plus informelle, et permet de capturer en temps réel les émotions, les frustrations, les succès et les dynamiques interpersonnelles de l'équipe. Ce canal de communication joue un rôle fondamental en tant que source de données primaires pour l'analyse. Les messages échangés dans cet espace contiennent une mine d'informations précieuses sur l'avancement du projet, les défis rencontrés, les besoins en ressources, ainsi que l'état d'esprit des membres de l'équipe. Les discussions incluent souvent des mises à jour sur l'état des tâches, des questions techniques, des demandes d'assistance et des retours sur le

travail effectué, offrant une vue en temps réel sur la progression du projet. Le chat est particulièrement utile pour suivre l'évolution des tâches et des projets de manière dynamique, permettant de voir non seulement ce qui fonctionne bien, mais aussi où se trouvent les obstacles et les blocages. Les messages échangés offrent un aperçu immédiat des moments où les membres de l'équipe rencontrent des difficultés ou ont besoin de soutien, ainsi que des périodes de collaboration accrue et de progrès rapide. De cette manière, le chat capture une image plus précise et actuelle des interactions au sein de l'équipe par rapport à d'autres canaux de communication. C'est à partir de ce flux continu d'échanges que l'outil d'analyse recueille les données nécessaires pour effectuer une analyse approfondie. Les messages sont systématiquement passés au crible pour identifier des indicateurs de performance tels que la fréquence des échanges, la tonalité des messages, et les schémas récurrents de communication. Des signaux d'alerte potentiels peuvent être détectés en temps réel, tels que des périodes de faible communication ou des discussions marquant un blocage dans l'avancement du projet. Ainsi, cette étape de collecte de données est essentielle car chaque message, chaque question, et chaque réponse contribue à former une mosaïque complexe des dynamiques de l'équipe. En analysant ces interactions, il est possible de comprendre le climat général du projet, d'identifier les tensions potentielles et de fournir aux décideurs des informations critiques pour intervenir efficacement. Sans cette étape, il serait impossible de capter avec autant de précision les interactions en temps réel et d'avoir une vision complète et nuancée du déroulement du projet.

### **3. Formatage de données**

Les données sont enregistrées et formatées dans un fichier .csv de manière à créer un « scénario » qui peut ensuite être chargé dans l'outil logiciel. Une fois que les données sont collectées à partir du canal de chat, elles doivent être structurées de manière cohérente pour faciliter leur traitement et leur analyse. Cette étape cruciale consiste à transférer ces données dans un format standardisé, généralement un fichier CSV (Comma-Separated Values). Le CSV est un format de fichier largement adopté en raison de sa simplicité et de sa compatibilité avec de nombreux outils d'analyse de données et de visualisation. Ce formatage est essentiel pour transformer les messages de chat non structurés en données structurées prêtes à être

traitées automatiquement par des algorithmes et des outils d'analyse. Pendant cette phase, chaque message de chat est extrait du flux de communication, puis structuré dans le fichier CSV avec plusieurs colonnes spécifiques. Parmi ces colonnes figurent des informations clés telles que l'horodatage, qui permet de situer précisément chaque message dans le temps ; l'identifiant de l'auteur, qui associe chaque message à son expéditeur ; et bien entendu, le contenu du message lui-même. En plus de ces données de base, des métadonnées supplémentaires peuvent également être incluses dans certaines colonnes, telles que des étiquettes de sentiment (positif, neutre ou négatif), ou encore des catégories d'interaction, comme « demande d'aide », « résolution de problème » ou « avancement de la tâche ». Le formatage des données sous forme de fichier CSV joue un rôle fondamental dans le processus d'analyse. En structurant les données de cette manière, on garantit non seulement leur compatibilité avec divers outils d'analyse, mais on facilite également leurs tri, filtrage et organisation. Par exemple, une fois les données organisées en colonnes, il devient plus simple de les segmenter selon des critères spécifiques, comme la date, l'auteur, ou même le type d'interaction. Cela permet une analyse plus ciblée et plus efficace, en isolant les données pertinentes pour répondre à des questions spécifiques, comme la détection des périodes de blocage ou l'analyse de la collaboration au sein de l'équipe. De plus, en utilisant un format standardisé comme le CSV, on garantit l'interopérabilité avec d'autres systèmes et outils. Les données peuvent ainsi être facilement partagées entre différentes plateformes d'analyse ou intégrées dans d'autres processus analytiques. Cela ouvre également la possibilité de réutiliser les données dans divers contextes, comme l'analyse historique des performances ou la création de rapports sur l'évolution des dynamiques de l'équipe au fil du temps. Cette étape de formatage prépare les données brutes pour l'analyse en les organisant de manière cohérente et exploitable. Elle constitue une phase clé dans le processus d'analyse, garantissant que les informations extraites du canal de chat sont prêtes à être traitées efficacement, tout en assurant leur compatibilité et leur flexibilité pour des usages futurs.

#### **4. Analyses NLP**

On applique la chaîne présentée précédemment : Tokenization, analyse des sentiments, NER, parsing syntaxique. Calcul des métriques : Calcul des fréquences des mots-clés, des

scores de sentiments, et des relations d'entités. L'outil d'analyse est le cœur du processus, jouant un rôle fondamental dans la transformation des données brutes en informations exploitables pour les décideurs. Une fois que les données de chat ont été correctement formatées, elles sont soumises à une série d'algorithmes, spécifiquement conçus pour extraire des insights significatifs à partir des échanges entre les membres de l'équipe. Cet outil s'appuie sur des techniques connues en traitement automatique du langage naturel (NLP) pour analyser et interpréter le contenu des messages, dans le but de fournir une compréhension détaillée et nuancée des dynamiques au sein de l'équipe. L'une des principales fonctionnalités de cet outil est l'analyse des sentiments. En parcourant les messages de chat, l'outil est capable d'évaluer le ton émotionnel des communications. Chaque message est classé comme étant de nature positive, négative ou neutre. Cette analyse des sentiments permet d'obtenir une image claire de l'humeur générale de l'équipe, et d'identifier les moments où des tensions ou des frustrations peuvent apparaître, ou au contraire, des périodes dont les membres de l'équipe, expriment un sentiment de satisfaction et d'accomplissement. Outre l'analyse des sentiments, l'outil intègre également des techniques de reconnaissance d'entités nommées (NER). Cette approche permet d'identifier automatiquement les éléments clés mentionnés dans les conversations, tels que les noms des membres de l'équipe, les dates importantes, les noms des projets ou des sous-tâches spécifiques. Cette capacité à extraire des entités est cruciale pour comprendre le contexte des discussions et repérer rapidement les sujets ou les individus au centre des préoccupations de l'équipe. En parallèle, l'outil est programmé pour détecter des mots-clés spécifiques, notamment ceux liés aux blocages, aux demandes d'assistance ou aux succès marquants. Par exemple, des mots ou des phrases comme « bloqué », « urgent », « problème » ou « résolu » sont automatiquement repérés et signalés. Cela permet d'alerter rapidement les gestionnaires de projet ou le Scrum Master sur des points nécessitant une intervention immédiate ou une attention particulière, garantissant ainsi que les obstacles potentiels ne sont pas négligés. Les résultats de cette analyse sont ensuite condensés sous forme de métriques et de rapports détaillés. Ces rapports incluent des informations telles que la fréquence des interactions entre les membres de l'équipe, les périodes où l'équipe a connu des tensions ou des moments de forte collaboration, et le niveau de satisfaction global des membres. En examinant ces métriques, les gestionnaires peuvent

non seulement suivre l'avancement du projet, mais aussi prendre des décisions éclairées basées sur des données objectives et concrètes. Ainsi, l'outil d'analyse joue permet l'extraction d'informations exploitables à partir des données de chat brutes. Grâce à ses capacités d'analyse des sentiments, de reconnaissance d'entités et de détection de mots-clés, il fournit une vue claire et objective de l'état de santé du projet et de l'équipe. En offrant aux gestionnaires des indicateurs clairs et des rapports détaillés, cet outil permet de prendre des décisions éclairées et proactives, contribuant ainsi à la réussite du projet en anticipant les problèmes et en optimisant la collaboration au sein de l'équipe.

## **5. Tableaux de bord**

Génération des graphiques des tendances et des alertes potentielles pour le gestionnaire de projet. Création des tableaux de bord. Le module de visualisation constitue la dernière étape du processus d'analyse de données. Une fois que l'outil d'analyse a extrait et traité les informations pertinentes issues des échanges de l'équipe, ces résultats doivent être présentés de manière claire, intuitive et accessible aux décideurs, tels que le chef de projet ou le Scrum Master. C'est ici que le module de visualisation entre en jeu. Son rôle principal est de transformer les données analytiques brutes en représentations graphiques et visuelles facilement compréhensibles, facilitant ainsi l'interprétation des résultats par les utilisateurs. Ce module est essentiel car il permet de synthétiser des informations complexes en un format visuel qui aide les utilisateurs à comprendre rapidement les dynamiques de l'équipe, ainsi que l'état d'avancement du projet. Plutôt que de naviguer dans des rapports textuels ou des tableaux de données, les décideurs ont accès à des graphiques et à des visualisations interactives qui leur fournissent une vue d'ensemble immédiate des points clés. Par exemple, des graphiques à barres peuvent illustrer la répartition des tâches au sein de l'équipe, en montrant qui est responsable de quoi et dans quelle mesure chaque tâche a été accomplie. Ces graphiques permettent également de repérer les goulots d'étranglement potentiels ou les membres surchargés. Des courbes de tendances, quant à elles, peuvent démontrer l'évolution des sentiments des membres de l'équipe au fil du temps, en visualisant des fluctuations émotionnelles telles que la satisfaction ou la frustration. Cela permet aux gestionnaires de mieux comprendre l'ambiance générale de l'équipe et d'anticiper les moments de tension ou

de surmenage. En outre, des jauges ou des diagrammes circulaires peuvent être utilisés pour montrer les progrès réalisés par rapport aux objectifs fixés. Ces types de visualisations aident à quantifier la distance restant à parcourir pour atteindre un jalon, ou pour évaluer dans quelle mesure les efforts de l'équipe ont permis de se rapprocher des résultats attendus. Ce type de représentation est particulièrement utile pour suivre les KPI (indicateurs clés de performance) et pour s'assurer que l'équipe reste alignée sur les objectifs stratégiques du projet. La visualisation des données joue également un rôle déterminant dans l'identification de schémas récurrents et d'anomalies qui ne seraient pas immédiatement visibles dans des rapports textuels ou des tableaux de données brutes. Par exemple, une augmentation soudaine des messages concernant des blocages peut être repérée instantanément grâce à une augmentation dans une courbe ou à un pic dans un graphique en barres. Cela permet une détection plus rapide des problèmes potentiels et, par conséquent, une intervention plus efficace pour résoudre ces problèmes. Le tableau de bord interactif du module de visualisation offre également une grande flexibilité, en permettant aux utilisateurs de filtrer les données selon divers critères, tels que la période, le type de tâche, ou encore les membres de l'équipe concernés. Cette fonctionnalité offre aux décideurs une vue d'ensemble personnalisée qui répond à leurs besoins spécifiques et leur permet de se concentrer sur les informations les plus pertinentes pour leur rôle. Par exemple, un chef de projet peut filtrer les données pour ne voir que les interactions et les progrès réalisés sur une tâche critique, tandis qu'un Scrum Master peut se concentrer sur l'analyse des dynamiques interpersonnelles au sein de l'équipe. En somme, cette étape permet de rendre les résultats de l'analyse non seulement exploitables, mais également faciles à interpréter et à utiliser pour la prise de décision. Grâce au module de visualisation, les décideurs pourront prendre des décisions plus rapides et mieux informées, car les données sont présentées de manière claire, visuelle et personnalisable, permettant ainsi d'agir en temps opportun pour maximiser l'efficacité et le succès du projet.

## **6. Visualisation des outils d'aide à la décision**

Le gestionnaire du projet consulte les tableaux de bord générés à l'étape précédente et utilise les informations pour prendre des décisions et résoudre des conflits. Ces tableaux de

bord sont conçus pour synthétiser les données complexes en visualisations claires et compréhensibles, permettant ainsi au gestionnaire d'identifier rapidement les points critiques nécessitant une attention particulière. L'efficacité de ces outils repose sur leur capacité à transformer des données brutes en informations exploitables, aidant ainsi le gestionnaire à non seulement suivre l'avancement du projet, mais aussi à anticiper et à atténuer les risques. La visualisation joue ici un rôle crucial en permettant une interprétation rapide et précise des données, facilitant ainsi des décisions plus stratégiques et proactives.

### **3.3 EXPÉRIMENTATION ET RÉSULTATS**

Dans cette section, nous présentons l'expérimentation réalisée avec notre outil d'analyse de langage naturel (NLP), spécialement conçu pour évaluer les interactions entre les membres d'une équipe Scrum en utilisant les données de clavardage. L'objectif principal de cette expérimentation était de tester, de façon sommaire, la capacité de l'outil à fournir une analyse utile des communications au sein de l'équipe en identifiant les tendances émergentes, en détectant les points de blocage potentiels et en analysant les sentiments exprimés par les membres.

L'outil a été appliqué à des données provenant de conversations simulées entre les différents membres d'une équipe Scrum, y compris des développeurs, un Scrum Master ou chef de projet, un designer et des testeurs. Ces simulations ont été construites de manière à représenter des scénarios réalistes dans le cadre d'un projet Scrum, couvrant les phases de planification des sprints, les réunions de revue de sprint, ainsi que des discussions autour de la résolution de problèmes et des blocages. Les échanges de clavardage incluaient des discussions sur la répartition des tâches, des questions techniques, des demandes d'assistance et des retours sur les progrès accomplis. Notre outil NLP utilise plusieurs techniques d'analyse, notamment l'analyse des sentiments pour évaluer l'état émotionnel des membres de l'équipe et détecter les moments de stress ou de satisfaction. En plus de cela, des algorithmes de reconnaissance d'entités nommées (NER) ont été utilisés pour identifier les projets, tâches, et personnes mentionnées dans les discussions, permettant de mieux comprendre le contexte des échanges. La combinaison de ces techniques a permis à notre

outil de cartographier les dynamiques de l'équipe, d'identifier les périodes de collaboration intensive ou de tension, et de fournir des indicateurs clés sur les performances et les problèmes internes de l'équipe.

### **3.3.1 Protocole expérimental**

L'expérimentation a été menée en utilisant des scénarios simulés d'échanges conversationnels, élaborés à l'aide de Chat GPT. Ces simulations visaient à reproduire des interactions réalistes entre les membres d'une équipe Scrum impliquée dans un projet de développement de jeux vidéo. L'objectif principal de cette expérimentation était d'évaluer les capacités de notre outil d'analyse de langage naturel (NLP) à identifier les dynamiques de communication, à repérer les points de blocage au sein du projet, ainsi qu'à analyser l'état émotionnel global des membres de l'équipe au cours du processus. Les scénarios simulés ont été conçus pour refléter différentes phases et situations courantes d'un projet Scrum. Ils incluaient, par exemple, des discussions sur la planification des sprints, la résolution des problèmes techniques, la répartition des tâches, ainsi que des retours réguliers sur l'avancement du projet. Ces conversations ont impliqué divers rôles au sein de l'équipe, notamment les développeurs, le Scrum Master, le designer, et les testeurs. Chacun de ces rôles a contribué à enrichir les échanges, en apportant des perspectives différentes sur les défis rencontrés dans le projet. Grâce à l'utilisation de Chat GPT pour générer ces scénarios, nous avons pu recréer des discussions variées, tant formelles qu'informelles, avec un degré de réalisme élevé. Les interactions simulées ont permis de tester la robustesse de notre outil dans différents contextes, tels que des situations de crise (comme un retard important dans le développement d'une fonctionnalité clé) ou des périodes de succès (lorsqu'un jalon important a été atteint sans difficulté majeure). Les données issues de ces simulations ont été analysées par notre outil pour détecter des schémas de communication récurrents, tels que des périodes de forte collaboration ou, au contraire, des moments de tension et de blocage. L'analyse des sentiments a également été utilisée pour déterminer les fluctuations émotionnelles au sein de l'équipe, offrant ainsi une vue globale sur la santé du projet et l'état d'esprit des membres de l'équipe.

### 3.3.2 Configuration de l'expérimentation

Dans cette section, nous allons décrire plus en détail la configuration de notre expérimentation.

**Nombre de simulations** : 4 simulations distinctes ont été créées, chacune représentant une phase différente d'un projet Scrum (exemple : planification du sprint, réunion de revue de sprint, résolution d'un blocage, etc.). Nombre de messages par simulation : Chaque simulation contenait entre 200 et 300 messages échangés entre les membres de l'équipe sur une période simulée de deux semaines. Au total, c'est 1200 messages générés à travers 4 scénarios distincts.

**Participants simulés** : Chaque simulation comportait 6 à 8 membres d'équipe, incluant un Scrum Master, un Désigner un développeur et un Testeur.

**Scénarios simulés** : Les échanges incluaient des messages liés à la planification des tâches, la gestion des priorités, les blocages rencontrés, et les réactions émotionnelles à des changements dans le projet.

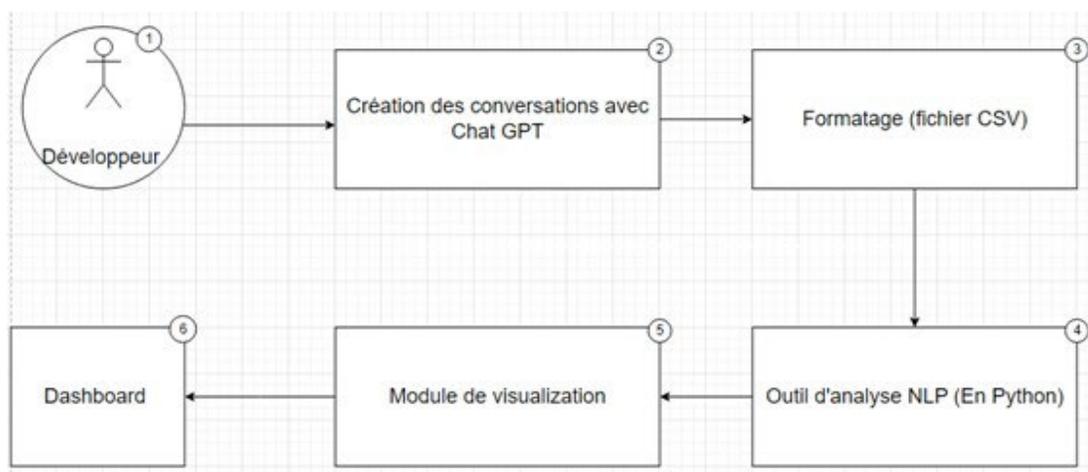


Figure 3 Architecture de l'environnement de simulation pour les tests

### 3.3.3 Protocole expérimental

Les simulations ont été réalisées selon un protocole standardisé, tel que décrit à la figure 3. De façon sommaire, la chaîne d'expérimentation se déroule comme suit. 1. Le

développeur de l’outil créé les paramètres du scénario. 2. Les paramètres sont entrés dans Chat GPT (version 4) pour effectuer la génération des conversations et simuler des échanges réalistes basés sur ces scénarios Scrum courants. 3. Les échanges générés sont ensuite passés à l’outil de formatage, pour simuler l’extraction à partir d’une plateforme de chat comme Microsoft Teams. 4. Ensuite, le passage à l’outil d’analyse NLP afin d’effectuer le traitement des messages pour en extraire les émotions, les entités nommées, les points de blocage, et d’autres métriques pertinentes. 5. Collecte des résultats et création des tableaux de bord à l’aide du module de visualisation. Les résultats de l’analyse NLP ont été recueillis sous forme de rapports statistiques pour générer les visualisations graphiques. 6. Le développeur visualise les tableaux générés pour analyser la qualité du résultat en fonction du scénario.

### **3.3.4 Résultats obtenus**

**Analyse des sentiments :** L’outil a détecté des variations significatives dans l’état émotionnel des membres de l’équipe tout au long des simulations. Par exemple, durant les phases de blocage (problèmes techniques non résolus), une augmentation notable des sentiments négatifs a été observée. Inversement, lors des réunions de planification réussies, les sentiments étaient majoritairement positifs.

**Points de blocage :** L’outil a réussi à identifier plusieurs points de blocage critiques en détectant des mots-clés comme « bloqué », « problème » ou « urgent », associés à des entités spécifiques (noms de tâches ou membres de l’équipe). Ces points de blocage ont été mis en évidence dans les rapports, offrant ainsi au gestionnaire une vue d’ensemble des obstacles à surmonter.

**Visualisations des tendances :** Les résultats ont été visualisés sous forme de graphiques montrant l’évolution des émotions au fil du temps, les moments clés où des blocages ont été détectés, ainsi que la fréquence des interactions entre certains membres de l’équipe (analyse des réseaux de communication). Un des graphiques montre l’évolution des sentiments positifs et négatifs au cours des deux semaines de simulation. On y observe des pics de sentiments négatifs lors de l’apparition de blocages, suivis d’une reprise des

sentiments positifs après la résolution de ces problèmes. Vous trouveriez ici jointes les différentes phases évolutives du projet.

### **3.3.5 Phase de planification**

La phase de planification, comme illustrée par l'analyse NLP de l'image 4, est une étape critique où l'équipe de projet se réunit pour définir les objectifs et les priorités des tâches à accomplir. Cette phase implique généralement des discussions approfondies entre les membres de l'équipe, y compris le chef de projet, les développeurs, les designers, et les testeurs, pour organiser les prochaines étapes du projet. Dans l'image, trois types d'analyses ont été sélectionnés : analyse de sentiments, reconnaissance d'entités, et analyse syntaxique. Ces analyses permettent d'évaluer la qualité des échanges pendant la phase de planification. Le résultat montre un score très élevé, avec une évaluation « Excellent » et 5 étoiles, indiquant que la communication au sein de l'équipe a été efficace et positive. Cela signifie que les discussions ont été constructives, les rôles bien définis, et les priorités clairement établies, sans conflits majeurs. Les résultats graphiques montrent également une constance dans les scores élevés (proches de 1.00), renforçant l'idée que la phase de planification s'est déroulée de manière fluide et harmonieuse. L'équipe semble avoir réussi à établir un consensus sur les objectifs et les tâches à accomplir, ce qui pose les bases d'une bonne exécution du projet dans les phases suivantes.

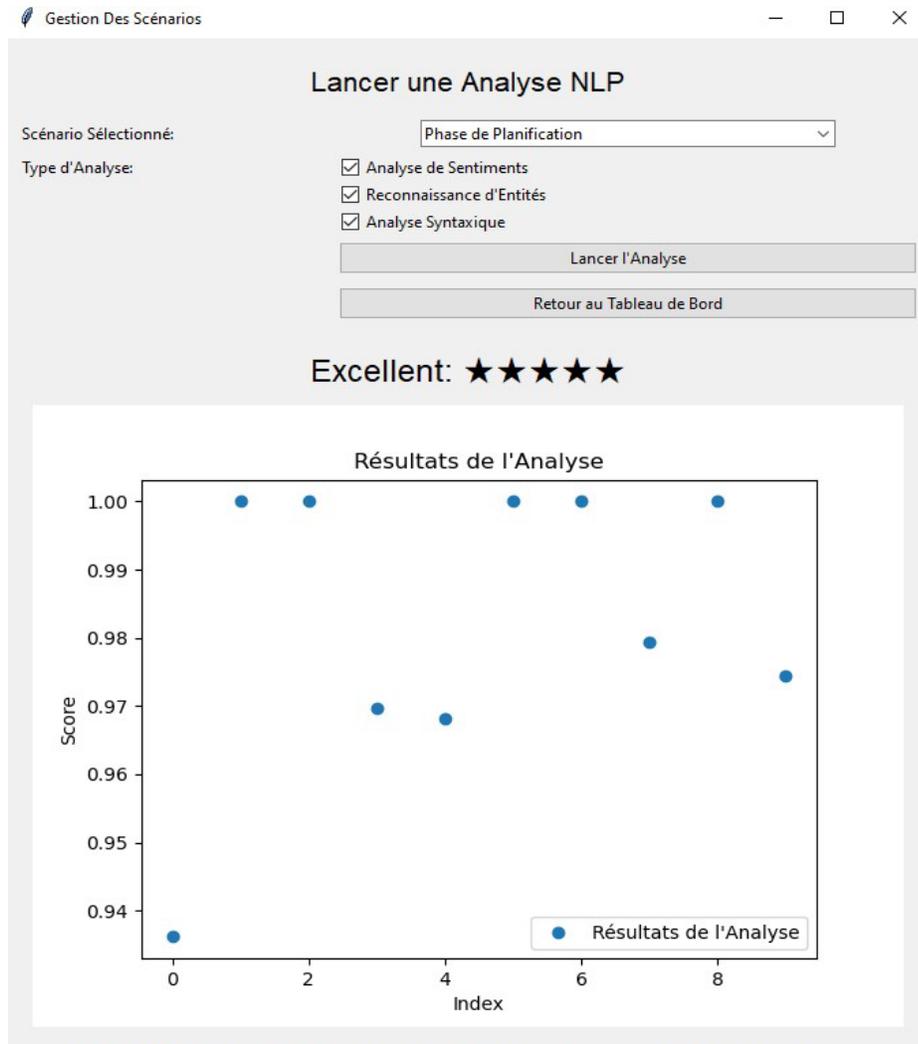


Figure 4 Résultats de l'analyse de la phase de planification

### 3.3.6 Phase de développement

Dans la phase de développement, l'équipe de projet travaille sur l'implémentation des fonctionnalités définies lors de la phase de planification. Cette étape implique une collaboration étroite entre les développeurs, les testeurs, les designers et le chef de projet pour s'assurer que le projet progresse comme prévu et que les fonctionnalités sont correctement développées. Sur l'image 5, l'analyse NLP montre une évaluation de cette phase avec une note « Excellent » de 5 étoiles, indiquant que les échanges au sein de l'équipe se sont déroulés de manière efficace et positive.



*Figure 5 Phases de développement*

Trois types d'analyses ont été appliquées : analyse de sentiments, reconnaissance d'entités, et analyse syntaxique, ce qui permet d'avoir une vue d'ensemble complète des interactions de l'équipe. Les résultats du graphique montrent une dispersion légèrement plus grande par rapport à la phase de planification, mais les scores restent très élevés (entre 0.92 et 1.00). Cela pourrait indiquer que, bien que la phase de développement soit plus complexe avec potentiellement plus de défis techniques, l'équipe reste cohérente et communique efficacement. L'excellente performance dans cette phase peut être attribuée à la bonne coordination des tâches, à la résolution rapide des problèmes rencontrés, et à une collaboration fluide entre les membres de l'équipe. Cette dynamique positive favorise une avancée harmonieuse du projet vers les étapes suivantes.

### **3.3.7 Phase de développement et correction des bugs**

L'image 6 présente les résultats d'une analyse NLP appliquée à la phase de développement et correction de bugs d'un projet, avec une évaluation indiquant un conflit marqué par une note de 2 étoiles sur 5. Trois types d'analyses ont été réalisées : analyse de sentiments, reconnaissance d'entités, et analyse syntaxique. Ces outils ont permis de détecter une baisse significative de la qualité des interactions au sein de l'équipe, avec des scores relativement bas sur l'échelle graphique, allant de 0 à 0,7. Cette phase est particulièrement critique, car la pression pour corriger les bugs avant la livraison peut créer des tensions entre les membres de l'équipe. Les résultats montrent des indices de désaccords ou de communication inefficace, ce qui pourrait expliquer les scores variables et parfois très faibles observés dans le graphique. La mention « Conflit » avec deux étoiles indique que les discussions entre les membres du projet ont probablement souffert d'une mauvaise coordination ou de malentendus, menant à une baisse de la productivité et à des tensions dans le groupe. Ces résultats montrent qu'il est essentiel de résoudre ces conflits pour améliorer la performance de l'équipe et éviter des retards ou des erreurs dans les phases suivantes du projet.

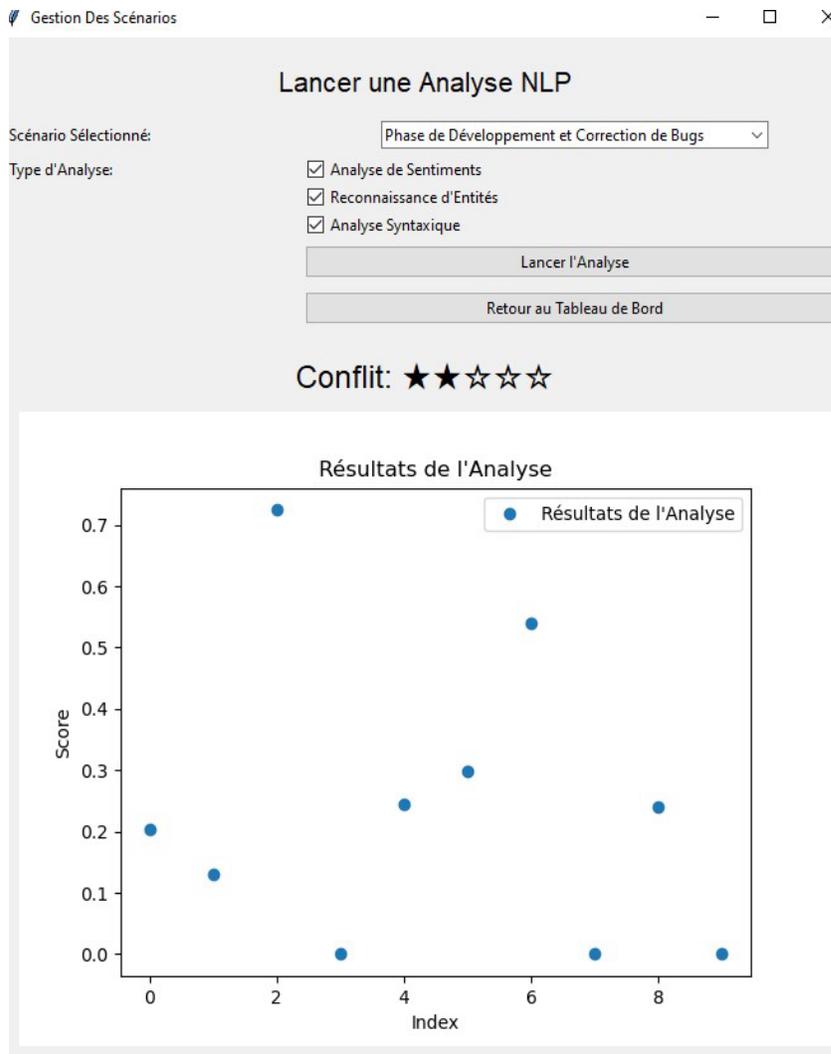


Figure 6 Phases de développement et correction des bugs

### 3.3.8 Phases de test et de livraison

L'image 7 montre une analyse NLP appliquée à la phase de test et de livraison d'un projet, avec une évaluation de la santé communicationnelle de l'équipe marquée par 3 étoiles sur 5. Cette évaluation montre que, bien que la communication soit globalement positive, certains problèmes persistent. Les trois types d'analyses appliquées, à savoir l'analyse de sentiments, la reconnaissance d'entités, et l'analyse syntaxique, offrent une vue complète des échanges entre les membres de l'équipe pendant cette phase importante.

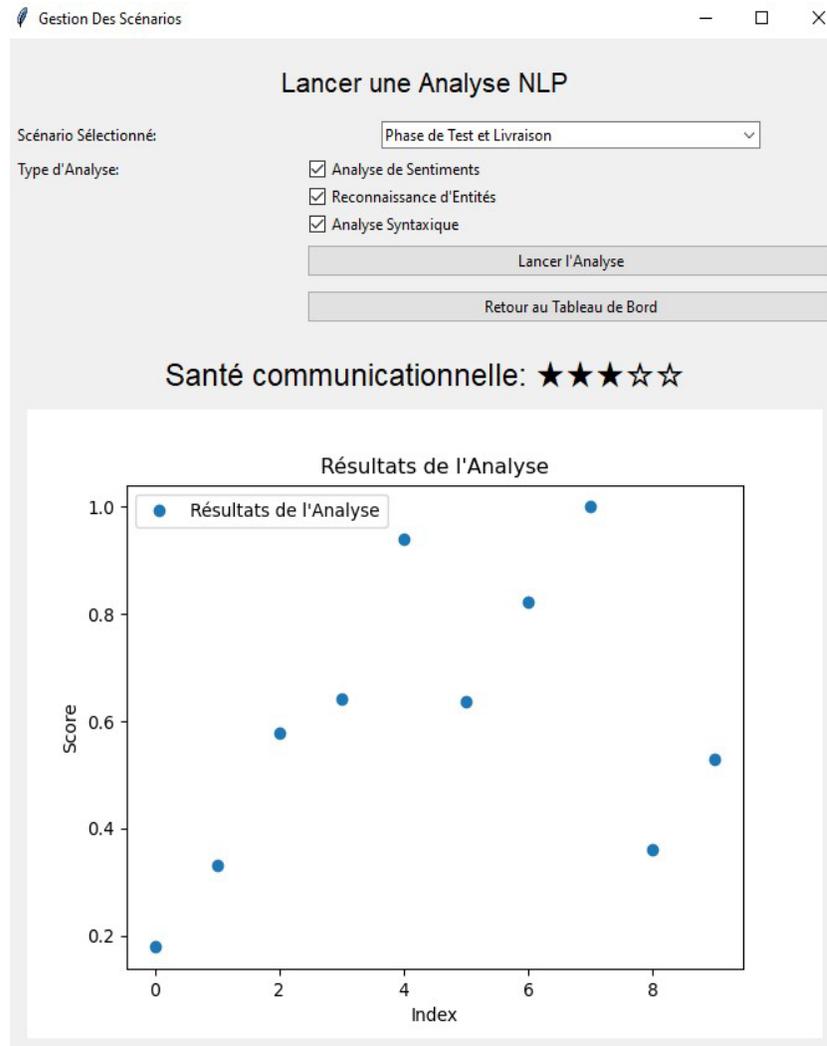


Figure 7 Phase de test et livraison

Les résultats graphiques montrent une certaine variabilité, avec des scores allant de 0,2 à 1,0, ce qui reflète une communication mixte. Certains moments des échanges ont probablement été positifs et constructifs, tandis que d'autres ont pu être marqués par des difficultés ou des tensions. Cette phase, étant souvent critique à l'approche de la livraison, peut être stressante pour les équipes. Les membres sont sous pression pour corriger les derniers bugs, tester les fonctionnalités, et livrer le produit dans les délais impartis. Le score de 3 étoiles suggère que, bien que la majorité des discussions aient été productives, des problèmes de communication ou des désaccords ont pu émerger lors des tests finaux et de la préparation de la livraison. Ces problèmes peuvent inclure des désaccords sur les priorités de test, des difficultés techniques imprévues, ou des malentendus concernant la livraison. En

conclusion, l'analyse montre que la santé communicationnelle est globalement acceptable, mais que l'équipe pourrait bénéficier d'une meilleure coordination et d'une meilleure gestion des conflits pour améliorer cette dernière phase avant la livraison finale du projet.

### **3.4 VALEUR AJOUTÉE DE L'OUTIL**

Sans l'outil, le gestionnaire de projet devrait lire manuellement chaque message pour identifier les problèmes, les sentiments exprimés, et les blocages. Cette approche est non seulement chronophage, mais aussi sujette aux erreurs humaines et aux biais d'interprétation. Notre outil, en revanche, permet une détection automatique des émotions et des blocages. Il identifie instantanément les problèmes potentiels dans la communication et alerte le gestionnaire sur les éléments nécessitant une intervention. Il offre également une analyse des tendances. Il est capable de détecter des tendances émotionnelles sur une période donnée, ce qui permet d'anticiper des baisses de motivation ou des conflits potentiels au sein de l'équipe. Enfin, il offre une vue d'ensemble de la communication de l'équipe. Grâce à la reconnaissance des entités nommées, il permet d'identifier les personnes ou les tâches associées à des problèmes spécifiques, facilitant ainsi la prise de décisions ciblées.

### **3.5 CONCLUSION**

Ce chapitre a présenté un nouvel outil logiciel basé sur le traitement automatique du langage naturel (NLP) conçu pour aider à la gestion de projets dans un cadre Scrum, en particulier pour les équipes travaillant à distance. L'outil exploite les échanges de clavardage des membres de l'équipe pour fournir une analyse approfondie des dynamiques de communication, des sentiments exprimés et des blocages rencontrés. L'expérimentation a démontré la capacité de cet outil à détecter automatiquement les problèmes de communication, à suivre l'évolution des émotions au sein de l'équipe, et à fournir des visualisations claires et exploitables pour les gestionnaires de projet. Grâce à des techniques de NLP telles que l'analyse des sentiments et la reconnaissance d'entités nommées, notre outil permet aux gestionnaires de mieux comprendre la santé de la communication dans leur équipe et d'intervenir de manière proactive pour résoudre des problèmes avant qu'ils ne

s'aggravent. Ce logiciel représente une avancée significative dans la gestion de projet à distance en fournissant un moyen d'automatiser la surveillance de la communication et en offrant des informations exploitables pour améliorer la productivité et la collaboration. L'outil offre un avantage considérable par rapport aux méthodes manuelles, permettant aux gestionnaires de mieux piloter les équipes Scrum dans des environnements numériques et dispersés.

## **CHAPITRE 4**

### **CONCLUSION ET PERSPECTIVE**

L'évolution rapide de l'industrie des technologies de l'information et la croissance exponentielle des projets complexes, notamment dans le secteur des jeux vidéo, soulignent l'importance croissante des approches adaptées pour la gestion de projets. Les équipes interdisciplinaires, souvent géographiquement dispersées et fonctionnant à distance, rencontrent des défis uniques qui nécessitent des solutions innovantes. L'application du cadre Scrum, traditionnellement conçu pour des équipes colocalisées, révèle des lacunes lorsqu'il est transposé à un environnement de télétravail. Les problèmes de communication et de confiance, essentiels à la réussite de toute équipe agile, sont particulièrement accentués dans ce contexte virtuel.

Ce mémoire a abordé ces défis en proposant une solution technologique basée sur l'analyse du langage naturel (NLP). En exploitant les outils de communication privilégiés des équipes distantes, à savoir les plateformes de chat, nous avons développé un module informatique capable d'analyser les échanges textuels pour évaluer la santé de la communication au sein de l'équipe. Ce module, implémenté en Python, utilise des techniques de NLP pour extraire des données pertinentes des interactions quotidiennes et appliquer une analyse quantitative afin de fournir des feedbacks constructifs aux gestionnaires de projet.

L'outil proposé a été testé avec des scénarios de communication générés par Chat GPT, ce qui a permis d'évaluer son efficacité dans des conditions contrôlées. Les résultats de ces tests indiquent que le module est capable de détecter des signaux de communication problématiques et de fournir des recommandations pertinentes pour améliorer la dynamique de l'équipe.

Cette approche offre une perspective nouvelle sur la manière dont les outils numériques peuvent compléter et renforcer les pratiques de gestion de projet Scrum, en particulier dans des contextes de travail à distance.

L'intégration de la NLP dans la gestion de projets Scrum représente une avancée significative vers une meilleure compréhension et la gestion des interactions au sein des équipes virtuelles. En facilitant une analyse approfondie des communications, cet outil aide non seulement à identifier les problèmes émergents mais aussi à intervenir de manière proactive pour maintenir une collaboration efficace. Il permet ainsi de répondre aux défis actuels du télétravail et d'améliorer les processus de gestion de projet dans un environnement numérique en constante évolution.

Néanmoins, cette recherche ouvre également la voie à plusieurs pistes de réflexion et d'amélioration. Les futurs travaux pourraient explorer l'intégration de fonctionnalités supplémentaires, telles que la détection automatique des conflits ou l'évaluation en temps réel des dynamiques de groupe, pour enrichir encore davantage les capacités de l'outil. Il serait également pertinent de mener des études supplémentaires dans des environnements de travail réels pour valider l'efficacité du module dans des situations variées et pour affiner les modèles d'analyse utilisés.

En conclusion, ce mémoire contribue à l'avancement des pratiques de gestion de projet en offrant une solution innovante aux défis posés par le télétravail. L'application de la NLP dans ce contexte montre un potentiel prometteur pour améliorer la communication et la collaboration au sein des équipes Scrum distantes, et ouvre la voie à de nouvelles opportunités de recherche et d'innovation dans la gestion de projets agiles.

## BIBLIOGRAPHIE

- [1] Gouvernement du Canada. Profil du secteur Canadien des Technologies de l'information et des communication, <https://ised-isde.canada.ca/site/technologies-numeriques-tic/fr/profil><https://ised-isde.canada.ca/site/technologies-numeriques-tic/fr/profil-secteur-canadien-tic>, 2021.
- [2] Statista, Work from home: remote & hybrid work - Statistics & Facts, <https://www.statista.com/topics/6565/work-from-home-and-remote><https://www.statista.com/topics/6565/work-from-home-and-remote-work/work/#topicOverview> , 2024.
- [3] Bojan Grebic & Aleksandra Stojanović, Application of the Scrum Framework on Projects in IT sector, European Project Management Journal, 11(2):37-46, 2022.
- [4] K. Schwaber and J. Sutherland, “The scrum guide,” 2020, 14p.
- [5] B. N. M. Yusuf, “Communications and trust is a key factor to success in virtual teams collaborations,” International Journal of Business and Technopreneurship, vol. 2, no. 3, pp. 399–413, 2012.
- [6] M. Takkunen, “Scrum implementation in a virtual team environment,” 2014. [Online]. Available: <https://www.theseus.fi/handle/10024/81863>
- [7] B. J. Gamage, « The impact of project management in virtual environment: A software industry perspective, » arXiv:1601.01220, 2016.
- [8] Khurana, D., Koli, A., Khatter, K. et al. Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl 82, 3713–3744 (2023).
- [9] By Jan Zizka, Frantisek Darena, Arnost Svoboda, Text Mining with Machine Learning Principles and Techniques, CRC Press, 368 pages, 2020.
- [10] M. Balasubramani, Ramya Maranan, Dr. Manesh R. Palav, Dr. Archana Ravindra Salve, Kamarajugadda Tulasi, Vigneswara Rao, Dr. Raja Ambethkar, Construction Project Management Using Natural Language Processing Technology, International Conferences on Contemporary Computing and Informatics, 2023.
- [11] G. Giuda, Mirko Locatelli, Marco Schievano, L. Pellegrini, Giulia Pattini, Paolo Ettore Giana, E. Seghezzi, Natural Language Processing for Information and Project Management , Journal of Digital Transformation of the Design, Construction and Management Processes of the Built Environment, 2019.
- [12] Leszek Ziora, Natural Language Processing in the Support of Business Organization Management Intelligent Systems with Applications, 2021.

- [13] I. K. Raharjana, D. Siahaan, Chastine Fatichah, User Stories and Natural Language Processing: A Systematic Literature Review, IEEE Access, 2021
- [14] Kaleemunnisa, Christelle Scharff, Krishna Mohan Bathula, and Kaiyin Chen. 2023. Analyzing Scrum Team Impediments Using NLP. In *Frontiers in Software Engineering Education: Second International Workshop, FISEE 2023, Villebrumier, France, January 23–25, 2023*.
- [15] B. N. M. Yusuf, “Communications and trust is a key factor to success in virtual teams collaborations,” *International Journal of Business and Technopreneurship*, vol. 2, no. 3, pp. 399–413, 2012.
- [16] R. J. Morrison-Smith S., “Challenges and barriers in virtual teams: a literature review,” in *SN Applied Sciences*, 2020, pp. 1–33.
- [17] L. Nankap et al., “Adapting the scrum framework to the needs of virtual teams of game developers with multi-site members,” in *ACM ICSIM 2022 Conf. on Software Eng. and Inf. Man. ACM*, 2022, pp. 19–25.
- [18] Edmondson, A., Psychological safety and learning behavior in work teams. *Administrative science quarterly*, 1999. 44(2): p. 350-383.
- [19] Lee, J. and K. Toutanova, Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3(8).
- [20] Vaswani, A., et al., Attention is all you need. *Advances in neural information processing systems*, 2017. 30.
- [21] Katada, S., S. Okada, and K. Komatani, Effects of physiological signals in different types of multimodal sentiment estimation. *IEEE Transactions on Affective Computing*, 2022. 14(3): p. 2443-2457.
- [22] Yadav, V. and S. Bethard, A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.
- [23] Hovy, D. and S.L. Spruit. The social impact of natural language processing. in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016.
- [24] S. Ghai, S. Lakhanpal, B. Ramola, R. R. M. Al-Tae and M. B. Alazzam, "Natural Language Processing in Solving Resource Constrained Project Scheduling Problems," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 256-260.
- [25] Hussain et al., Natural Language Processing for Construction Management: A Literature Review, *Construction Research Congress 2024*, 607-618, 2024.

- [26] Kang, Y., Cai, Z., Tan, C. W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, 7(2), 139–172.
- [27] Beck, K., et al. (2001) The Agile Manifesto. Agile Alliance. <http://agilemanifesto.org/>
- [28] Petersen, Kai; Wohlin, Claes; Baca, Dejan (2009). "The Waterfall Model in Large-Scale Development". In Bomarius, Frank; Oivo, Markku; Jaring, Päivi; Abrahamsson, Pekka (eds.). *Product-Focused Software Process Improvement. Lecture Notes in Business Information Processing*. Vol. 32. Berlin, Heidelberg: Springer. pp. 386–400.
- [29] Hamzane I. et al.: MDA Transformation Process from predictive project management methodologies to agile project management methodologies, *Journal of Emerging Trends in Engineering Research* 8 (9), 2020.
- [30] Ademola Fathia, AI in Agile Development: Enhancing Scrum and Kanban Methodologies, publié uniquement sur ResearchGate, 2024, 16 pages.
- [31] Y. K. Dwivedi, N. Kshetri, L. Hughes, E. L. Slade, A. Jeyaraj, A. K. Kar, et al., "'So what if ChatGPT wrote it?'" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *International Journal of Information Management*, vol. 71, p. 102642, 2023.
- [32] B. Khadka, « Data analysis theory and practice: Case: Python and Excel Tools, » 2019.
- [33] D. L. Hansen, B. Shneiderman, and M. Smith, "Visualizing threaded conversation networks: mining message boards and email lists for actionable insights," in *International Conference on Active Media Technology*, pp. 47-62, Berlin, Heidelberg: Springer Berlin Heidelberg, Aug. 2010.
- [34] D. Goldenberg, "Social network analysis: From graph theory to applications with Python," *arXiv preprint arXiv:2102.10014*, 2021.
- [35] Y. Sastri and A. J. Kuttyamma, "NetworkX and Matplotlib: An Analysis," *Int. J. Sci. Eng. Res*, vol. 4, pp. 1-4, 2013.
- [36] T. McKenzie, M. M. Trujillo, and S. Hoermann, "Software engineering practices and methods in the game development industry," in *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pp. 181-193, October 2019.
- [37] Rico D.F., *The Paradox of Agile Project Management and Virtual Teams*, davidfrico.com, 2019.

- [38] C. M. Chubala, A. Hunter, L. Dithurbide, et H. F. Neyedli, "Building Situation Awareness and Team Cohesion through Effective Information Sharing in a Distributed Team-Based Command and Control Scenario," in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 67, no. 1, pp. 710-715, Sep. 2023.
- [39] I. Benke, M. T. Knierim, et A. Maedche, "Chatbot-based emotion management for distributed teams: A participatory design study," Proceedings of the ACM on Human-Computer Interaction, vol. 4, no. CSCW2, pp. 1-30, 2020.
- [40] V. S. A. Retna, P. Brundha, and G. RajKumar, "People's Behaviour analysis in chat message using natural language processing," in 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Feb. 2021, pp. 1128-1133.
- [41] A. Srivastav, G. Indira, S. A. Tiwaskar, A. Garg, S. Singh, et S. Das, "Text Mining for Intelligent Information Processing and Analysis," in 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Dec. 2023, pp. 1-5.
- [42] S. V. Joshua, S. Singh, J. Krithika, S. S. Pund, N. Kulkarni, et S. K. Chaturvedi, "Exploring the Potential of Text Mining for Automated Information Analysis," in 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Dec. 2023, pp. 1-6.
- [43] J. Klünder, J. Horstmann, and O. Karras, "Identifying the mood of a software development team by analyzing text-based communication in chats with machine learning," in Human-Centered Software Engineering: 8th IFIP WG 13.2 International Working Conference, HCSE 2020, Eindhoven, The Netherlands, November 30–December 2, 2020, Proceedings 8, Springer International Publishing, pp. 133-151, 2020.
- [44] K. Xie, G. Di Tosto, L. Lu, and Y. S. Cho, "Detecting leadership in peer-moderated online collaborative learning through text mining and social network analysis," The Internet and Higher Education, vol. 38, pp. 9-17, 2018.
- [45] S. J. White, "Conversation analysis: an introduction to methodology, data collection, and analysis," 2019.
- [46] R. Sujatha et K. Nimala, "Text-based Conversation Analysis Techniques on Social Media using Statistical methods," in 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Jan. 2022, pp. 1-11.

## APPENDICE : EXEMPLES DE SCÉNARIOS DU PROJET

### Jour 1 : Lundi matin

- CP : "Bonjour à tous ! Nous devons finaliser la planification du sprint. DEV, peux-tu t'occuper du module d'authentification ?"
- DEV : "Oui, bien sûr, je m'en occupe. Est-ce que tu as des spécifications détaillées à partager, DES ?"
- DES : « Oui, je viens de terminer les maquettes pour l'écran d'authentification. Je les envoie sur le Slack dans quelques minutes. »
- TEST : "Je vais préparer des scénarios de test pour vérifier la sécurité et la performance."
- CP : "Super, on doit aussi discuter du tableau de bord principal. DES, peux-tu préparer les maquettes ?"
- DES : « D'accord, j'aurai besoin de précisions sur les fonctionnalités. »

### Jour 2 : Mardi après-midi

- CP : « Nous devons aussi planifier le développement des nouvelles fonctionnalités pour le tableau de bord. »
- DEV : "Je vais commencer par l'intégration des nouvelles API. DES, tu pourras me fournir les icônes nécessaires ?"
- DES : « Oui, je vais m'en occuper après avoir terminé les maquettes. »
- TEST : « Je vais préparer les tests unitaires pour les nouvelles fonctionnalités. »
- CP : « Merci à tous pour votre collaboration. On se retrouve demain pour faire un point. »

### Jour 3 : Mercredi matin

- CP : « Le sprint est lancé, comment ça se passe ? »
- DEV : "Je suis en train d'intégrer les API pour le tableau de bord. Je rencontre quelques soucis avec l'authentification."
- DES : « Je travaille sur les icônes et les maquettes. Je vais les envoyer dès qu'elles seront prêtes. »
- TEST : « Je vais commencer à tester l'authentification dès que DEV aura terminé. »
- CP : « Super. Assurez-vous de bien documenter les problèmes que vous rencontrez. »

#### **Jour 4 : Jeudi après-midi**

- DEV : « L'intégration des API est presque terminée, mais j'ai encore des problèmes avec l'authentification. »
- DES : « Les icônes sont prêtes. DEV, je les ai mises sur Slack. »
- CP : « On doit s'assurer que l'authentification fonctionne avant de continuer. »
- TEST : « Je vais vérifier ça ce soir et vous ferez un retour demain matin »

#### **Jour 6 : Lundi matin**

- CP : « Les tests montrent quelques bugs sur l'authentification. DEV, peux-tu t'en occuper aujourd'hui ? »
- DEV : « Oui, je vais m'en occuper tout de suite. TEST, tu pourras retester après la correction ? »
- TEST : « Bien sûr. Je vais aussi tester les nouvelles fonctionnalités du tableau de bord. »
- DES : « Je vais vérifier que les modifications n'ont pas affecté le design. »
- CP : « Merci, c'est important qu'on corrige tout avant la fin de la semaine. »

#### **Jour 7 : Mardi après-midi**

- DEV : « J'ai corrigé les bugs, tout devrait fonctionner correctement maintenant. »
- TEST : « Je viens de finir les tests, il reste un petit problème avec la validation des courriels. »
- CP : « Nous devons corriger cela avant la fin de la semaine pour rester dans les délais. »
- DEV : « Je vais m'en occuper tout de suite. »

#### **Jour 8 : Mercredi matin**

- CP : « Nous entrons en phase de test final. DEV, DES, avez-vous des retours de dernière minute avant de livrer ? »
- DEV : « Tout semble bon de mon côté. J'attends juste la confirmation des tests finaux. »
- TEST : « Je vais terminer les tests aujourd'hui. J'ai déjà validé la majorité des fonctionnalités. »
- DES : « Je vais revoir une dernière fois les maquettes pour m'assurer qu'elles correspondent bien à ce qui a été livré. »

### **Jour 9 : Jeudi après-midi**

- CP : « Nous sommes prêts pour la livraison demain. Assurez-vous que tout est bien documenté. »
- TEST : « Je vais préparer le rapport final de test. »
- DEV : « Je m'assure que le code est bien nettoyé et que tout est prêt pour la mise en production. »
- DES : « Je vais faire un dernier contrôle des éléments graphiques. »

### **Jour 10 : Vendredi matin**

- CP : « Félicitations à tous ! Nous avons livré avec succès. Merci pour votre excellent travail tout au long de ces deux semaines. »
- DEV : « C'était une bonne expérience. Content que tout se soit bien passé. »
- TEST : « Ravi que nos tests aient pu contribuer à une livraison réussie. »
- DES : « Le produit final est à la hauteur de ce que nous avons imaginé. Bravo à tous ! »