UQAC
Université du Québec
à Chicoutimi

# GÉNÉRATION ET AFFINEMENT AUTOMATISÉS D'HISTOIRES UTILISATEURS AGILES À PARTIR DE DOCUMENTS DE CONCEPTION (GDD) DE JEUX VIDÉO À L'AIDE DE MODÈLES DE LANGAGE À GRANDE TAILLE

**PAR WAIL KHELIFA**

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

**QUÉBEC, CANADA**

## RÉSUMÉ

Les *Game Design Document* (GDD) constituent une référence centrale dans le développement de jeux vidéo, capturant les mécaniques de jeu, les éléments narratifs et les objectifs d'expérience utilisateur. Bien qu'indispensables pour orienter la production, ces documents sont souvent très peu structurés et riches en narration, ce qui rend difficile l'extraction d'artefacts agiles tels que les *user stories* de manière cohérente et efficace.

Cette étude propose une méthodologie novatrice exploitant les *Large Language Models* (LLM) et l'ingénierie de prompts afin d'automatiser la génération et l'amélioration de *user stories* directement à partir des GDD. Trois stratégies de prompting : *Chain-of-Thought* (CoT), *Refine-and-Thought* (RaT) et une variante hybride ont été appliquées à deux LLM (ChatGPT et DeepSeek), aboutissant à six chaînes de génération distinctes par GDD. Des centaines de *user stories* ont été produites à partir d'un corpus hétérogène de huit GDD (quatre commerciaux et quatre académiques) et évaluées systématiquement à l'aide d'AQUSA, un cadre d'évaluation automatisé fondé sur des critères de qualité agiles formels.

L'évaluation initiale a révélé des défauts fréquents liés au minimalisme, à l'atomicité et à l'uniformité. Cependant, l'intégration directe des directives d'AQUSA dans les prompts a permis d'éliminer entièrement ces lacunes, démontrant l'efficacité d'une boucle de raffinement guidée par des standards agiles formels. Au-delà des améliorations de qualité, les résultats soulignent la faisabilité et la scalabilité (*passage à l'échelle*) de l'intégration de la génération de *user stories* basée sur les LLM dans les flux de travail agiles, réduisant considérablement l'effort manuel, améliorant la traçabilité et comblant le fossé entre la conception narrative et les exigences d'ingénierie dans le développement de jeux modernes.

D'un point de vue scientifique, ce travail apporte une méthodologie reproductible combinant ingénierie de prompts, évaluation automatisée et raffinement itératif. D'un point de vue pratique, il ouvre des perspectives prometteuses d'intégration industrielle dans des outils de gestion de projet tels que Jira ou GitLab. Ses principes sont également transférables à d'autres domaines où des documents narratifs non structurés doivent être transformés en exigences opérationnelles, comme l'éducation, la formation ou les *serious games*.

## TABLE DES MATIÈRES

## LISTE DES TABLEAUX

## LISTE DES FIGURES

## REMERCIEMENTS

Ce mémoire n'aurait pas été possible sans le soutien indéfectible de mes parents, qui m'ont toujours encouragé tout au long de mon parcours. J'aimerais exprimer ma profonde gratitude envers mes professeurs, M. Bruno Bouchard et M. Gilles Imbeau, pour leur accompagnement, leurs conseils avisés et leur disponibilité.

Je souhaite également remercier toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail, collègues, amis, ainsi que toutes celles et ceux qui m'ont offert leur aide et leur soutien au fil de ce projet.

À toutes ces personnes, je témoigne ma sincère reconnaissance pour leur rôle dans mon cheminement personnel et professionnel.

# CHAPITRE I

# INTRODUCTION

## 1.1 PANORAMA DE L'INDUSTRIE DU JEU VIDÉO (2023–2025)

L'industrie du jeu vidéo occupe aujourd'hui une place importante dans l'économie mondiale du divertissement. Selon les rapports récents de l'ESA et de Newzoo (ESA); Newzoo (2023), le marché mondial du jeu vidéo a été estimé à environ 184 milliards USD en 2023, et plusieurs analyses anticipent qu'il pourrait atteindre près de 200 milliards USD à l'horizon 2025.

Cette croissance est portée par l'essor des plateformes numériques, l'évolution rapide des technologies matérielles et logicielles, ainsi qu'une demande accrue pour des expériences interactives immersives.

Sur le plan des plateformes, la répartition des revenus illustre un changement structurel majeur : le segment mobile domine désormais, représentant près de 50 % du marché mondial (ESA). Cette prédominance s'explique par l'accessibilité des smartphones et par la diffusion des modèles économiques *free-to-play* et des microtransactions, qui soutiennent une adoption massive et durable Hamari *et al.* (2017).

La cadence de production témoigne d'un environnement extrêmement compétitif. Chaque mois, des centaines de nouveaux titres paraissent sur les principales plateformes de distribution numérique (*Steam*, *PlayStation Store*, *App Store*). Cette abondance accroît à la fois la diversité de l'offre et la pression exercée sur les studios pour se différencier par l'innovation technologique, narrative et artistique Paavilainen *et al.* (2020). Parallèlement, l'échelle des

projets s'accroît, avec des équipes pluridisciplinaires, des budgets excédant fréquemment ceux de superproductions cinématographiques et des cycles de développement pluriannuels, ainsi que l'intégration de technologies émergentes (réalité virtuelle, simulation en temps réel, intelligence artificielle adaptative), ce qui accentue la complexité des productions Tiedemann *et al.* (2024).

Dans ce contexte, l'ingénierie documentaire et la gestion rigoureuse des processus de développement deviennent déterminantes pour la compétitivité. La capacité à structurer, tracer et exploiter efficacement l'information de conception est d'autant plus critique que les cycles de vie se réduisent et que les attentes des joueurs évoluent rapidement. D'où l'intérêt croissant pour des méthodes et outils d'automatisation afin de rationaliser la documentation dans l'industrie vidéoludique.

## 1.2 CADENCE DE PUBLICATION ET COMPLEXITÉ CROISSANTE DES PRODUCTIONS

Un marqueur saillant de l'industrie contemporaine réside dans le volume croissant de nouvelles sorties. Selon les données compilées par *SteamDB* et d'autres observateurs du secteur Zhang *et al.* (2023), plus de 12 000 jeux auraient été publiés sur la plateforme *Steam* en 2022, soit près d'un millier par mois. Les estimations récentes suggèrent un volume comparable pour les années 2023 et 2024, illustrant un rythme de production particulièrement soutenu. Cette tendance reflète la démocratisation des outils de développement, la multiplication des studios indépendants et l'intensification de la concurrence sur les marchés numériques.

Parallèlement, la complexité des productions majeures s'est considérablement accrue. Des succès récents tels que *The Legend of Zelda : Tears of the Kingdom* (2023) et *Baldur's Gate 3* (2023) illustrent des cycles de développement pluriannuels, une coordination interna-

tionale et l'intégration poussée de technologies avancées Tiedemann *et al.* (2024). Au-delà du segment AAA, les studios intermédiaires et indépendants opèrent désormais dans des environnements logiciels de plus en plus sophistiqués, avec des pipelines de production hétérogènes et des exigences accrues en matière de qualité et de support post-lancement Paavilainen *et al.* (2020). La montée en puissance du modèle *games-as-a-service* prolonge la durée de vie des jeux via des mises à jour continues, extensions et microtransactions, imposant ainsi des pratiques de *live-ops* de plus en plus rigoureuses.

Dans ce contexte, la gestion des artefacts de conception devient critique non seulement lors de la phase initiale de développement, mais tout au long du cycle de vie du produit. Cela inclut la version des GDD, la traçabilité des décisions, et l'alignement continu entre les extraits du GDD et les éléments du *Product Backlog*. Les cadences de publication et les opérations récurrentes (correctifs, patch notes, équilibrages) soulignent le besoin de mécanismes capables d'extraire, de structurer et de maintenir à jour les artefacts de développement à grande échelle. C'est à ce besoin opérationnel que répond l'exploration d'approches d'automatisation fondées sur les modèles de langage.

## 1.3 LE GAME DESIGN DOCUMENT (GDD) : DÉFINITION, HISTORIQUE, RÔLE ET CONTENU

Le *Game design document* (GDD) représente l'un des artefacts les plus caractéristiques du développement vidéoludique. Apparu dans les années 1980–1990, il a progressivement pris le rôle de document de référence pour organiser, formaliser et communiquer la vision de jeu auprès des différentes parties prenantes Fullerton (2014). Avec la professionnalisation et l'industrialisation du secteur, le GDD s'est diffusé comme un standard de facto de la documentation de conception, bien qu'aucune norme universelle n'ait jamais été établie

Salas & Togelius (2021).

Un GDD vise à capturer de manière exhaustive les informations conceptuelles et techniques nécessaires à la création d'un jeu. Ses principales composantes incluent :

— **Mécaniques de gameplay** : règles, systèmes d'interaction, progression et équilibrage des niveaux Perera *et al.* (2023).

— **Narration et univers** : scénario, dialogues, personnages et contexte diégétique.

— **Direction artistique** : styles graphiques, palettes de couleurs, design sonore et musique.

— **Structure technique** : architecture logicielle, plateformes cibles, contraintes de performance et exigences matérielles.

— **Interfaces et ergonomie** : navigation des menus, *heads-up display* (HUD) et schémas de contrôle.

Au-delà de la description, le GDD joue un rôle central de coordination et de communication : il sert à la fois de *plan directeur* pour la mise en œuvre technique et de *pont* entre les équipes aux expertises variées (conception, programmation, art, production, test) Rahman *et al.* (2024). En ce sens, il soutient une compréhension partagée et cohérente de la vision du projet.

Un GDD bien structuré et régulièrement mis à jour réduit les ambiguïtés, limite les malentendus et facilite la traçabilité des décisions de conception. Il favorise également la reproductibilité et la réutilisation des pratiques, en constituant une mémoire documentaire réutilisable pour des projets ultérieurs Heger *et al.* (2024). Toutefois, la forte variabilité des formats et l'absence d'un standard universel compliquent le traitement automatique, un enjeu qui sera analysé plus en détail dans la section suivante.

## 1.4 VARIABILITÉ ET ABSENCE DE STANDARDISATION DU GDD

Malgré son rôle central, le *Game Design Document* (GDD) ne bénéficie d'aucun standard universel. Il n'existe ni format commun ni structure normaliséeMateus *et al.* (2023b). Chaque studio, voire chaque équipe, adopte ses propres conventions de granularité, d'organisation et de terminologie. Cette hétérogénéité se traduit par des écarts substantiels : certains GDD privilégient une approche narrative et descriptive, quand d'autres accentuent les dimensions techniques ou économiques.

Cette variabilité complique la communication inter-équipes et freine la mutualisation des pratiques documentaires. Elle constitue aussi un obstacle à l'automatisation de l'extraction et de l'analyse, les systèmes devant composer avec des structures dissemblables, des vocabulaires hétérogènes et des niveaux de détail inégaux Rahman *et al.* (2024). Concrètement, l'application directe de méthodes d'ingénierie des exigences reposant sur des formats stabilisés est limitée (p. ex. difficulté de *mapping* de schémas, repérage d'entités et alignement terme–concept, détection d'exigences implicites).

La spécificité du GDD apparaît plus nettement par contraste avec d'autres artefacts. Le *Product Requirements Document* (PRD), courant en ingénierie logicielle et en gestion de produits, est structuré autour de la définition d'exigences fonctionnelles et non-fonctionnelles selon des modèles souvent stabilisés en entreprise Lee & Kim (2020). De son côté, le *Business Process Model and Notation* (BPMN) fournit un langage graphique standardisé et interopérable entre différents outils et environnements de modélisation, permettant de représenter les processus métier tout en facilitant l'analyse, la simulation et l'automatisation.

## 1.5 AGILITÉ, SCRUM ET RÔLE DES USER STORIES

Face à l'accélération des cycles de développement et à la complexification des projets, les méthodologies agiles se sont imposées comme paradigme dominant dans l'industrie logicielle, y compris dans les studios de jeux vidéo Hoda (2020). Fondée sur quatre valeurs et douze principes du *Manifeste agile*, l'approche privilégie l'adaptabilité, la collaboration, la livraison incrémentale et l'attention continue aux besoins des utilisateurs finaux. Parmi ces cadres, *Scrum* s'est largement diffusé grâce à sa simplicité et à sa capacité à structurer le travail des équipes multidisciplinaires Schwaber & Sutherland (2020).

*Scrum* organise le développement en itérations courtes (*sprints*) de deux à quatre semaines, chacune visant la livraison d'un incrément fonctionnel. Il s'appuie sur un *Product Backlog* priorisé et évolutif, dont les éléments sont généralement exprimés sous forme de *user stories* Paternoster *et al.* (2014).

Une *user story* décrit de manière concise une fonctionnalité du point de vue d'un utilisateur ou d'un acteur système, en suivant le format *Connextra* : *« En tant que [acteur], je veux [fonctionnalité] afin de [bénéfice] »* Lucassen *et al.* (2016). Ce gabarit explicite la valeur métier et favorise une compréhension partagée entre parties prenantes, tout en soutenant la priorisation et la planification Tiedemann *et al.* (2024). Au-delà de la planification, les *user stories* servent de support de communication courant (développeurs, concepteurs, testeurs, responsables produit), pour s'assurer que les fonctionnalités livrées répondent aux besoins identifiés.

Dans le contexte vidéoludique, ces artefacts permettent d'articuler des éléments parfois abstraits (mécaniques de jeu, progression, immersion) sous une forme exploitable pour l'implémentation et le suivi.

## 1.6 PROBLÉMATIQUE DE RECHERCHE

Les constats précédents convergent vers un même point : les *Game design document* (GDD) sont des artefacts riches, volumineux et hétérogènes, rédigés en langage naturel, sans format commun Mateus *et al.* (2023b); Salas & Togelius (2021). La transformation de ces documents en *user stories* exploitables au sein d'un *Product Backlog* demeure largement manuelle et, par conséquent, coûteuse et fragile. Elle reste exposée à des biais d'interprétation et de subjectivité, à des ambiguïtés linguistiques et à une variabilité rédactionnelle qui nuisent à l'homogénéité des artefacts et à la reproductibilité des processus Lucassen *et al.* (2016); Rahman *et al.* (2024).

Cette situation pose un double défi. D'une part, elle limite la **scalabilité** des pratiques documentaires dans un contexte de cadence de publication élevée et de complexité croissante des productions. D'autre part, elle compromet la **traçabilité** et la **qualité** des exigences dérivées : l'absence de liens explicites entre extraits du GDD et *user stories* crée des zones d'incertitude sur l'origine des décisions de conception, tandis que l'hétérogénéité des formulations fragilise la planification agile et les activités de test Salas & Togelius (2021); Lucassen *et al.* (2016). En pratique, les studios arbitrent entre rapidité (génération à grande vitesse) et rigueur (clarté, atomicité, testabilité) : un compromis difficilement soutenable.

**Problématique** : *Comment transformer, de façon fiable et reproductible, des GDD narratifs et non standardisés en un ensemble de user stories de haute qualité, adaptées à une utilisation en contexte agile ?* Pour y répondre, il est nécessaire (i) de **réduire la dépendance à la rédaction manuelle** à l'aide de mécanismes d'automatisation capables d'absorber la variabilité structurelle et stylistique des GDD, (ii) de **contrôler la granularité** et la clarté des *user stories* générées afin d'en garantir l'actionnabilité, et (iii) d'**assurer une évaluation**

**objective et réplicable** à l'aide de cadres et d'outils dédiés (p. ex., QUS et AQUSA) Lucassen *et al.* (2016); Ferrari *et al.* (2017); Sharma *et al.* (2024); Tiedemann *et al.* (2024).

Ces exigences motivent l'exploration d'approches fondées sur les *Large Language Models* (LLM), dont les capacités récentes en compréhension et génération de texte laissent entrevoir la possibilité d'un pipeline GDD→*user stories* à la fois **extensible (scalable)** et **de qualité contrôlée** Rahman *et al.* (2024). Toutefois, ces modèles soulèvent des questions ouvertes (hallucinations, contrôle fin de la granularité, traçabilité des sources) qui imposent l'**intégration native de l'évaluation** dans la boucle de génération et de raffinement. La section suivante présente les pistes d'automatisation retenues (CoT, RaT, Hybride) et leur positionnement au regard de ces exigences.

## 1.7 AUTOMATISATION PAR LES MODÈLES DE LANGAGE : POTENTIEL ET DÉFIS

L'émergence des *Large Language Models* (LLM) tels que GPT-4 (ChatGPT) ou DeepSeek a profondément transformé les perspectives de l'ingénierie logicielle et des méthodes agiles. Leurs capacités en compréhension du langage naturel, en génération cohérente et en exécution d'instructions complexes ouvrent la voie à l'automatisation de tâches jusqu'ici manuelles, comme la génération de *user stories* à partir de documents textuels non structurés. Dans le contexte vidéoludique, les LLM présentent un potentiel particulier pour traiter la richesse descriptive et la variabilité des *Game design document* (GDD), en extrayant des fonctionnalités exploitables et en les traduisant en artefacts agiles structurés Rahman *et al.* (2024).

Cette automatisation vise à réduire le temps et l'effort cognitif nécessaires à la transformation de GDD volumineux en *Product Backlogs* initiaux, à limiter l'hétérogénéité issue de

la rédaction manuelle par l'application de gabarits reproductibles et à permettre le passage à l'échelle pour des corpus importants Heger *et al.* (2024).

Plusieurs stratégies de *prompting* sont mobilisées. Le *Chain-of-Thought* (CoT) guide un raisonnement explicite étape par étape jusqu'à la *user story* finale Wei *et al.* (2022). Le *Refine-and-Thought* (RaT) introduit une boucle de raffinement où les sorties initiales sont évaluées puis améliorées à l'aide de justifications successives Sharma *et al.* (2024). Une approche hybride combine ces deux logiques afin de tirer parti à la fois de la structuration du raisonnement et de la qualité issue du raffinement. Ces stratégies, prometteuses, requièrent toutefois une évaluation systématique de leur robustesse face à la diversité des GDD.

Des défis subsistent. Les modèles peuvent produire des sorties incohérentes ou des « hallucinations », c'est-à-dire des informations non présentes dans le document source Gilardi *et al.* (2023). La granularité des *user stories* demeure délicate à contrôler (trop générales et peu actionnables, ou au contraire excessivement détaillées). Enfin, la traçabilité source→artefact reste limitée (p. ex., rattachement explicite d'une *user story* à des extraits du GDD et à une version donnée), ce qui complique la validation et l'auditabilité. Ces limites imposent l'intégration de mécanismes d'évaluation et de contrôle qualité pour fiabiliser l'automatisation.

## 1.8 CADRES DE QUALITÉ : QUS ET AQUSA

L'automatisation de la génération de *user stories* n'est pertinente que si les artefacts produits satisfont des critères de qualité rigoureux. Une *user story* imprécise, redondante ou trop générale compromet la planification agile et la valeur livrée à l'utilisateur final. En réponse, plusieurs cadres formalisent la qualité des *user stories*.

Le *Quality User Story Framework* (QUS), proposé par Lucassen et al., constitue une

contribution majeure Lucassen *et al.* (2016). Il définit des critères tels que l'atomicité (une seule fonctionnalité par histoire), la clarté, l'uniformité de la formulation, l'absence d'ambiguïté et la testabilité. Ces dimensions, issues de l'ingénierie des exigences et des pratiques agiles, permettent d'objectiver une tâche jusque-là largement subjective.

Ces critères ont été opérationnalisés dans des outils d'évaluation automatique comme *AQUSA* (*Automated Quality User Story Analyzer*) Ferrari *et al.* (2017). *AQUSA* s'appuie sur des règles linguistiques et des patrons de détection pour repérer des défauts fréquents (formulations imprécises, omission d'acteur, redondances structurelles). En fournissant une analyse systématique et réplicable, l'outil dépasse les jugements subjectifs et soutient une amélioration incrémentale de la qualité des artefacts.

Dans le contexte de la génération automatique par LLM, l'intégration conjointe de **QUS** (référentiel de critères) et d'**AQUSA** (instrumentation automatisée) dans une boucle de génération et de raffinement est indispensable Sharma *et al.* (2024); Tiedemann *et al.* (2024). Elle permet de filtrer puis d'améliorer automatiquement les *user stories* afin d'assurer leur conformité aux standards agiles et leur exploitabilité dans des environnements réels.

## 1.9 OBJECTIFS, QUESTIONS DE RECHERCHE ET CONTRIBUTIONS

L'objectif principal de ce mémoire est de proposer et d'évaluer une approche automatisée pour générer des *user stories* de haute qualité à partir de *Game design document* (GDD), en s'appuyant sur des modèles de langage à grande échelle (LLM). L'ambition est de réduire la variabilité liée à la rédaction manuelle et d'assurer une qualité contrôlée des artefacts conformes aux pratiques agiles.

Plus précisément, ce travail poursuit trois objectifs complémentaires :

1. **Concevoir un pipeline de génération automatisée** intégrant des stratégies de *prompting* (CoT, RaT, Hybride) pour transformer un GDD en *Product Backlog* initial.
2. **Évaluer la qualité des *user stories*** à l'aide de cadres et d'outils établis **(QUS, AQUSA)** — clarté, atomicité, uniformité, testabilité.
3. **Comparer les stratégies et les modèles** (ChatGPT, DeepSeek) face à des GDD hétérogènes.

Ces objectifs se déclinent en questions de recherche :

— **RQ1** : Dans quelle mesure des LLM peuvent-ils générer automatiquement des *user stories* exploitables à partir de GDD variés selon les critères **QUS** et les analyses **AQUSA** ?

— **RQ2** : Quel est l'impact des stratégies de *prompting* (CoT, RaT, Hybride) sur la qualité (scores/défauts **AQUSA**) des *user stories* produites ?

— **RQ3** : Comment intégrer **AQUSA** dans une boucle de raffinement pour réduire les défauts détectés et améliorer les scores ?

— **RQ4** : Quelles différences observe-t-on entre LLM (ChatGPT, DeepSeek) sur des GDD académiques vs commerciaux ?

Les contributions attendues s'articulent autour de trois axes :

1. **Une méthodologie outillée** pour automatiser la transformation des GDD en artefacts agiles, intégrant des stratégies avancées de *prompting* et un mécanisme d'évaluation systématique.
2. **Un protocole expérimental comparatif** appliqué à un corpus de huit GDD (académiques et commerciaux), mettant en évidence l'influence des modèles et des stratégies de *prompting* sur la qualité des *user stories*.

3. **Une discussion critique et des pistes de recherche** pour améliorer l'automatisation documentaire en ingénierie logicielle, en particulier en contexte vidéoludique.

## 1.10 APERÇU EXPÉRIMENTAL ET ORGANISATION DU MÉMOIRE

L'approche proposée a été mise en œuvre et évaluée au moyen d'une étude expérimentale fondée sur un corpus de huit GDD, couvrant à la fois des projets académiques et des productions commerciales. Trois stratégies de *prompting* (CoT, RaT, Hybride) ont été appliquées à deux modèles de langage (ChatGPT, DeepSeek), aboutissant à **six** combinaisons expérimentales. Les *user stories* produites ont été évaluées selon les critères du *Quality User Story Framework* (AQUSA) et analysées automatiquement à l'aide de l'outil *AQUSA*, afin de mesurer leur clarté, leur atomicité, leur uniformité et leur testabilité. Cette démarche permet une comparaison systématique des stratégies et des modèles et met en évidence des leviers d'amélioration pour l'automatisation de la transformation des GDD en artefacts agiles.

Le reste de ce mémoire est structuré comme suit :

— Le **chapitre 2** présente l'état de l'art sur la génération automatique de *user stories* à partir de documents textuels, ainsi que sur l'amélioration de leur qualité à l'aide des grands modèles de langage (LLM).

— Le **chapitre 3** introduit et décrit en détail l'approche proposée pour la génération automatisée de *user stories* à partir de *Game design document*, en précisant les stratégies de *prompting*, le corpus utilisé et le protocole expérimental.

— Le **chapitre 4** présente les résultats obtenus à partir de l'évaluation expérimentale, en comparant les différentes approches et modèles selon les critères de qualité définis.

— Le **chapitre 5** discute les implications de ces résultats, analyse les menaces à la validité et propose des pistes d'amélioration et de recherche future.

— Enfin, le **chapitre 6** conclut le mémoire en résumant les principales contributions et en ouvrant sur les perspectives d'évolution du travail.

**Portée et limites.** Cette étude se concentre sur la transformation de GDD textuels en *user stories* et sur l'évaluation de leur qualité à l'aide de **QUS** (référentiel) et **AQUSA** (instrumentation). Elle n'aborde pas la génération d'assets, l'évaluation in situ auprès d'équipes industrielles, ni l'impact économique longitudinal des décisions de conception.

# CHAPITRE II

# TRAVAUX CONNEXES

Au cours des dernières années, l'intérêt académique pour l'automatisation de la documentation agile a fortement augmenté, en particulier avec l'essor du traitement automatique du langage naturel (NLP) et des grands modèles de langage (LLM), appliqués à la génération ou à l'amélioration de user stories Heger *et al.* (2024). Dans cette section, nous passons en revue les travaux antérieurs étroitement liés à notre recherche, que nous regroupons en deux grandes catégories : (1) les approches visant à générer automatiquement des user stories à partir de documents sources, et (2) les méthodes centrées sur l'évaluation ou l'amélioration de la qualité des user stories existantes à l'aide des LLM. À notre connaissance, aucune étude existante n'a encore abordé spécifiquement la génération automatique de user stories à partir de Game Design Document (GDD).

## 2.1 GÉNÉRATION AUTOMATIQUE DE USER STORIES À PARTIR DE DOCUMENTS

**Rahman et al. (2024)** introduisent *GeneUS*, un outil basé sur GPT-4 visant à automatiser la génération de *user stories* et de cas de test à partir de documents de spécifications logicielles Rahman *et al.* (2024). Cette étude s'inscrit dans le contexte de l'ingénierie des exigences (*Requirements Engineering*, RE), une phase particulièrement chronophage du cycle de développement logiciel où les besoins exprimés par les clients sont transformés en artefacts agiles exploitables. L'objectif principal de GeneUS est de réduire la charge cognitive et temporelle des ingénieurs en automatisant la conversion des spécifications en tâches unitaires intégrables dans des outils de gestion de projet (Jira, Azure DevOps, etc.).

Les données d'entrée utilisées pour l'évaluation sont composées de sept documents de spécifications de taille moyenne, six proviennent de cas d'étude classiques tirés du manuel de Sommerville (*Software Engineering*, 10e édition) et un est issu d'un projet industriel réel dans le domaine de la gestion d'événements. Tous les éléments non textuels (diagrammes, maquettes) ont été supprimés pour éviter les ambiguïtés lors du traitement par le modèle.

La méthodologie repose sur une stratégie de *prompting* originale, le *Refine-and-Thought* (RaT), conçue comme une variante du *Chain-of-Thought* (CoT). RaT opère en deux temps : (i) une phase de raffinement visant à nettoyer les symboles et informations redondantes extraites du document, puis (ii) une phase de raisonnement où le modèle produit des user stories structurées, enrichies de critères d'acceptation, de contraintes fonctionnelles et non fonctionnelles, ainsi que de cas de test associés. Cette approche enchaîne plusieurs blocs RaT afin d'assurer une génération plus cohérente et de limiter les effets d'« hallucination » observés avec les LLM classiques.

Les résultats expérimentaux montrent que l'outil est capable de produire des user stories détaillées et exploitables. Une évaluation empirique a été menée via le questionnaire *RUST* (Readability, Understandability, Specifiability, Technical aspects), administré à 50 praticiens issus de l'industrie logicielle et du milieu académique (expérience moyenne de 4 ans en ingénierie des exigences). Les réponses, notées sur une échelle de 1 à 5, indiquent une médiane de 4 pour la plupart des dimensions évaluées. Plus précisément, seuls 5 % des user stories manquent de détails techniques, 1 % présentent des ambiguïtés et 0,5 % sont jugées redondantes. Toutefois, certaines limites sont relevées : des faiblesses persistentent dans la spécificité des scénarios de test et dans la précision technique de certaines histoires, ce qui suggère un besoin d'amélioration pour les aspects fortement liés au contexte d'implémentation.

La principale contribution de GeneUS réside dans l'intégration d'une stratégie de *prompting* innovante pour l'automatisation partielle du RE, ouvrant la voie à ce que les auteurs appellent *Autoagile*, soit l'automatisation de bout en bout du processus agile grâce aux LLM. Néanmoins, l'étude reste limitée à des spécifications textuelles relativement structurées et n'a pas été testée sur des documents riches en narration comme les Game Design Document (GDD). De plus, l'évaluation repose sur un instrument (RUST) élaboré par les auteurs et non sur des cadres formels établis tels que INVEST ou AQUSA.

Par rapport à notre propre travail, GeneUS constitue une avancée majeure en démontrant la faisabilité d'une génération automatique de user stories et des cas de test de qualité acceptable à partir de documents RE standards. Cependant, notre recherche s'en distingue sur deux aspects fondamentaux : (i) l'application à des GDD, documents plus narratifs et hétérogènes que les spécifications traditionnelles, et (ii) l'intégration explicite de critères de qualité agile (AQUSA) directement dans la conception des prompts, afin de renforcer la rigueur, l'objectivité et la reproductibilité de l'évaluation.

**Raharjana et al. (2021)** ont conduit une revue systématique de la littérature portant sur l'application des techniques de traitement automatique du langage naturel (NLP) aux *user stories* Raharjana *et al.* (2021). Leur objectif était de dresser un état de l'art complet afin d'identifier les usages, les approches et les défis associés à l'intégration du NLP dans l'ingénierie des exigences agile. Pour ce faire, ils ont suivi le protocole PRISMA et ont interrogé six bases de données majeures (Scopus, IEEE Xplore, ACM DL, ScienceDirect, SpringerLink et Google Scholar), couvrant la période 2009—2020. À partir de 718 articles initiaux, un filtrage par critères d'inclusion/exclusion et une procédure de *snowballing* avant/arrière ont permis de retenir 38 études primaires pertinentes.

Les travaux identifiés ont été regroupés en quatre grandes catégories : (i) la détection de défauts dans les user stories (ambiguïtés, duplications, exigences incomplètes ou incohérentes), (ii) la génération automatique d'artefacts logiciels à partir de user stories (diagrammes UML, cas de test, modèles BPMN, extraits de code), (iii) l'identification d'abstractions clés et de modèles conceptuels (ontologies, modèles de buts, taxonomies de concepts), et (iv) la traçabilité entre user stories et autres artefacts logiciels (tests, code source, documents de conception). La prédominance des approches syntaxiques est manifeste : la majorité des études exploitent le *Part-of-Speech tagging*, le *dependency parsing* et des modèles vectoriels, tandis que les approches sémantiques et pragmatiques restent encore peu explorées.

L'analyse qualitative met en lumière plusieurs limites : seulement 26 % des études emploient des méthodes de validation rigoureuses (mesures de précision/rappel, comparaisons empiriques), et à peine 28 % se basent sur des travaux empiriques robustes plutôt que sur des propositions conceptuelles. De plus, la plupart des recherches sont réalisées en contexte académique avec des jeux de données restreints et hétérogènes, ce qui limite la reproductibilité et la généralisabilité des résultats. Les auteurs soulignent ainsi le besoin urgent de jeux de données ouverts et standardisés pour l'évaluation.

En termes de contribution, cette revue fournit une cartographie exhaustive des méthodes de NLP appliquées aux user stories et propose des pistes de recherche, notamment l'exploration des approches sémantiques et l'usage de l'apprentissage profond. Toutefois, elle montre aussi que très peu de travaux se sont intéressés à la génération complète et automatique de user stories au format standardisé (ex. Connextra) à partir de textes narratifs. Ce constat positionne directement notre propre recherche : alors que Raharjana et al. mettent en évidence l'immaturité des approches existantes et l'absence d'outils pour transformer des documents narratifs en artefacts agiles utilisables, notre travail vise à pallier cette lacune en mobilisant des LLM et

en intégrant des critères de qualité agile (AQUSA) pour produire et évaluer des user stories issues de Game Design Document (GDD).

**Nistala et al. (2022)** proposent une méthode de génération automatique de *user stories* adaptée aux pratiques agiles centrées sur les user stories (par ex. intégration dans Jira), en s'appuyant sur des documents techniques semi-structurés tels que les spécifications logicielles et les descriptions fonctionnelles Nistala *et al.* (2022). Leur objectif est de réduire la subjectivité et les divergences d'interprétation lors de la rédaction manuelle des user stories, tout en facilitant la transition entre les phases d'ingénierie des exigences et la planification agile.

Les données d'entrée sont constituées de documents issus de projets académiques et industriels, organisés en une hiérarchie de fonctionnalités (Feature Area, Major Feature, Sub Feature). La méthode s'appuie sur un système fondé sur des règles, combinant des techniques classiques de traitement automatique du langage naturel avec des heuristiques spécifiques au domaine logiciel. Concrètement, l'outil utilise la bibliothèque OpenNLP pour effectuer le *Part-of-Speech tagging* et l'analyse syntaxique, puis identifie les phrases exprimant des intentions fonctionnelles. Ces phrases sont ensuite reformulées en user stories selon le modèle Connextra (*« En tant que [acteur], je veux [fonctionnalité] afin de [bénéfice]. »*), enrichies de métadonnées contextuelles (module d'origine, dépendances fonctionnelles).

L'évaluation combine des métriques quantitatives (nombre de fonctionnalités et de user stories extraites, précision 94–98 %) et une validation qualitative par des experts, et repose sur des exemples illustratifs extraits des documents sources. Les auteurs mettent en avant une amélioration de la traçabilité entre les exigences initiales et les artefacts Scrum, ainsi qu'une meilleure cohérence contextuelle des user stories générées. Cependant, la démarche ne repose pas sur un cadre formel d'analyse de qualité tel qu'INVEST ou AQUSA, ce qui limite la validité externe des résultats. En outre, la dépendance forte à la structure syntaxique et à

l'organisation modulaire des documents restreint l'applicabilité de l'approche à des contextes bien balisés ; elle n'est pas adaptée à des documents non structurés ou narratifs comme les Game Design Document (GDD).

La contribution de ce travail réside dans la démonstration qu'un système basé sur des règles et des techniques de NLP classiques peut automatiser partiellement la transformation des spécifications en artefacts Scrum. Toutefois, comparée aux approches modernes exploitant les LLM, elle souffre d'un manque de flexibilité et de robustesse face à la variabilité des formats. Notre recherche s'inscrit dans la continuité de ce constat en cherchant à dépasser les limites des approches fondées sur des règles par l'utilisation de modèles de langage capables de traiter des documents narratifs complexes et en intégrant directement des critères de qualité agile dans le processus de génération.

**Mateus (2023)** propose dans son mémoire de maîtrise un cadre méthodologique visant à dériver automatiquement des *user stories* et des scénarios de test en langage Gherkin à partir de modèles *Business Process Model and Notation* (BPMN) Mateus *et al.* (2023a). L'étude s'inscrit dans le domaine de l'ingénierie des exigences, où l'un des défis majeurs est de garantir la traçabilité entre la modélisation des processus métiers et les artefacts agiles exploités dans le développement logiciel. L'objectif est de combler l'écart entre la représentation des processus (souvent utilisée pour analyser et optimiser les activités organisationnelles) et la spécification fonctionnelle sous forme de user stories, artefact central des méthodologies agiles.

La méthodologie repose sur des *patrons de transformation* systématiques : les acteurs des processus sont associés aux rôles dans les user stories, les tâches sont traduites en actions, et les passerelles (*gateways*) en comportements conditionnels. Ces patrons ont été intégrés dans un prototype développé en .NET (Visual Studio, MVC) et utilisant la bibliothèque *BPMN.io*, permettant d'analyser automatiquement des modèles BPMN (`.bpmn`) et de générer en sortie

des user stories textuelles au format Connextra, ainsi que des scénarios Gherkin exploitables pour le *Behaviour-Driven Development* (BDD).

L'évaluation a été réalisée à partir d'un ensemble de modèles BPMN, puis complétée par une validation auprès d'un groupe d'experts au moyen d'un questionnaire. Celui-ci portait notamment sur la couverture des transformations et sur la précision des extractions obtenues. Les résultats indiquent que l'approche contribue à améliorer la cohérence et la traçabilité entre les processus métiers et les exigences logicielles, tout en facilitant la communication entre analystes et développeurs. Toutefois, l'étude met en évidence plusieurs limites, notamment une dépendance à la qualité et au niveau de détail des modèles BPMN fournis, certaines imprécisions dans les extractions signalées par les experts, ainsi qu'une couverture incomplète de certains flux alternatifs ou cas plus complexes.

En termes de contribution, cette recherche illustre le potentiel des approches fondées sur des règles pour relier directement la modélisation des processus et les artefacts agiles. Elle constitue un jalon important pour l'automatisation des transformations documentaires, mais reste centrée sur des structures techniques diagrammatiques et standardisées. Par contraste, notre travail aborde le défi complémentaire de l'automatisation à partir de documents narratifs non structurés comme les Game Design Document (GDD). Ainsi, alors que l'approche de Mateus démontre la faisabilité de la conversion BPMN–user stories via des patrons explicites, notre recherche explore l'apport des modèles de langage (LLM) pour gérer la variabilité, l'ambiguïté et la richesse textuelle des documents de conception vidéoludique.

**Conclusion de la sous-section.** Les études analysées dans cette première catégorie proposent des approches diverses pour générer automatiquement des user stories à partir de documents techniques structurés. Elles s'appuient sur des pipelines de NLP classiques Nistala *et al.* (2022),

des règles de transformation Mateus *et al.* (2023a), des revues systématiques Raharjana *et al.* (2021), ou des stratégies de *prompting* avec les LLM Rahman *et al.* (2024). Toutefois, ces approches ciblent principalement des domaines comme la finance ou les systèmes d'entreprise, où les documents sources sont standardisés et bien structurés.

Aucune de ces méthodes n'a été conçue pour traiter les Game Design Document (GDD), qui sont narratifs, hétérogènes et souvent non structurés. De plus, les cadres formels d'évaluation de la qualité agile, tels qu'AQUSA ou INVEST, sont rarement intégrés. Notre recherche s'inscrit donc dans un espace encore peu exploré, en proposant une méthode guidée par des prompts enrichis de critères de qualité agile, adaptée aux spécificités des GDD et évaluée à l'aide du cadre AQUSA.

## 2.2 AMÉLIORATION OU ÉVALUATION DE LA QUALITÉ DES USER STORIES À L'AIDE DES LLM

Une seconde orientation de recherche s'intéresse non pas à la génération initiale de user stories, mais à leur amélioration ou à leur évaluation après rédaction. L'objectif est d'accroître leur qualité en termes de clarté, de cohérence et de conformité aux pratiques agiles.

**Zhang et al. (2024)** introduisent *ALAS* (*agile Language Agent System*), un système multi-agent basé sur des modèles de langage de grande taille (LLM), conçu pour améliorer la qualité des *user stories* existantes Zhang *et al.* (2024). L'étude part du constat que la rédaction et la révision de user stories demeurent fortement dépendantes de l'expertise des équipes, ce qui entraîne des variations de qualité et de cohérence. Pour répondre à ce problème, les auteurs proposent un cadre où chaque agent conversationnel incarne un rôle inspiré du *Product Owner* et de l'analyste des exigences. Ces agents collaborent à travers des prompts structurés et des

échanges simulés, afin de reproduire une revue collaborative telle qu'elle se déroule dans une équipe agile.

Les données d'entrée consistent en des user stories réelles, issues de six équipes industrielles, de l'entreprise Austrian Post Group IT. L'évaluation a mobilisé 11 praticiens expérimentés, incluant des *Product Owners*, des développeurs, des testeurs et un *Scrum Master*. Le protocole a consisté à soumettre un ensemble de user stories aux participants, avant et après amélioration par ALAS, et à mesurer la qualité perçue via un questionnaire aligné sur plusieurs critères INVEST (clarté, valeur métier, testabilité, granularité).

Les résultats montrent une amélioration notable de la clarté, de la cohérence et de la valeur métier perçue des user stories. Les versions générées par GPT-3.5 ont été jugées particulièrement lisibles et concises, tandis que celles produites par GPT-4 présentaient un contenu plus riche mais parfois excessivement détaillé, ce qui a nui à leur applicabilité immédiate dans un backlog agile. Ces observations mettent en évidence un compromis entre richesse descriptive et utilité pratique, selon le modèle employé.

Cependant, certaines limites sont soulignées. D'une part, l'échantillon évaluatif demeure restreint (11 participants, un seul contexte organisationnel), ce qui limite la généralisation des résultats. D'autre part, le système se concentre sur la révision de user stories déjà existantes, sans proposer un mécanisme complet de génération initiale à partir de documents sources complexes. Enfin, la dépendance à des modèles propriétaires (GPT-3.5, GPT-4) introduit une variabilité difficile à maîtriser et un risque de verbiage ou de sur-détail.

En termes de contribution, cette étude illustre de manière convaincante le potentiel des LLM pour simuler une revue collaborative et assister les équipes dans l'amélioration incrémentale des user stories. Elle met en lumière l'intérêt d'exploiter des dynamiques multi-

agents pour reproduire les échanges humains caractéristiques des pratiques agiles. Par contraste, notre travail se distingue en abordant la problématique en amont : plutôt que de corriger des stories déjà existantes, nous visons la génération automatique et contrôlée de user stories directement à partir de documents riches et non structurés tels que les *Game Design Document* (GDD). Cette différence de positionnement souligne la complémentarité entre approches orientées « amélioration » et approches orientées « génération ».

**Santos et al. (2025)** mènent une étude empirique intitulée *User Stories : Does ChatGPT Do It Better ?*, présentée à la conférence ICEIS 2025, qui vise à déterminer si ChatGPT peut produire des user stories de qualité comparable, voire supérieure, à celles rédigées par des praticiens humains dos Santos *et al.* (2025). Le travail s'appuie sur deux expériences successives, toutes deux évaluées selon le cadre QUS (Quality User Stories) proposé par Lucassen et al. (2016), qui analyse treize dimensions regroupées en aspects syntaxiques, sémantiques et pragmatiques.

Dans la première expérience, 30 étudiants en génie logiciel de l'Université Fédérale de l'Amazonas ont été divisés en groupes et invités à produire des user stories pour un système de gestion d'agence de voyages. Chaque groupe a d'abord rédigé des stories manuellement, puis en a généré une seconde série via ChatGPT en utilisant un prompt libre (*free form*). Au total, 126 user stories ont été collectées (62 manuelles et 64 générées automatiquement). L'analyse statistique révèle un taux moyen de conformité aux critères QUS de 64,28 % pour les stories manuelles et 64,23 % pour celles générées automatiquement, sans différence significative ($p = 0.983$). Toutefois, les stories générées par ChatGPT présentaient une plus grande homogénéité (écart-type 9,5 % contre 10,9 % pour les manuelles), ce qui suggère un potentiel de standardisation et de régularité dans l'écriture.

Face à ces résultats mitigés, les auteurs ont conçu une seconde expérience visant à tester une stratégie plus élaborée, dite *« Meta-Few-Shot Prompt »*. Celle-ci combine des instructions structurées intégrant explicitement les critères du cadre QUS (méta-prompting) avec des exemples concrets de bonnes et mauvaises stories (few-shot learning). L'application de ce prompt raffiné a permis de générer 15 nouvelles user stories, dont la qualité a été réévaluée. Les résultats montrent une amélioration substantielle : un taux de succès global de 88,57 % (contre 65,85 % pour les approches manuelle et libre), avec un gain particulièrement marqué sur les dimensions sémantiques (1,0 contre 0,87–0,90 précédemment) et pragmatiques (0,75 contre 0,68–0,69). L'amélioration est statistiquement significative ($p < 0.001$), confirmant l'efficacité de l'approche Meta-Few-Shot pour accroître la précision et la valeur des stories générées.

Cependant, des limites persistent : environ 40 % des user stories produites par la méthode raffinée souffraient encore d'un manque d'atomicité, ce qui affecte la clarté et la planification. De plus, la validité externe reste limitée, puisque les participants provenaient d'un contexte académique, et non de projets industriels. Enfin, les auteurs rappellent le risque d'hallucinations inhérent aux LLM, soulignant la nécessité d'une validation humaine des artefacts générés.

Cette étude met en évidence que la formulation du prompt constitue un facteur décisif pour la qualité des user stories produites par ChatGPT. Elle se rapproche de notre travail par son objectif d'améliorer la qualité des artefacts agiles grâce aux LLM. Toutefois, alors que Santos et al. se concentrent sur des contextes académiques génériques et sur l'évaluation directe de prompts, notre recherche vise à appliquer et affiner ces techniques dans un contexte narratif spécifique : les Game Design Document (GDD) en intégrant des mécanismes d'amélioration systématique via des critères comme AQUSA.

**da Silva Neo et al. (2024)** introduisent *User Story Tutor* (UST), un outil interactif en ligne visant à accompagner les praticiens agiles dans l'amélioration de la qualité de leurs user stories da Silva Neo *et al.* (2024). Ce travail s'inscrit dans un contexte où la rédaction de user stories souffre fréquemment de problèmes de clarté, de manque de structure et de non-respect des formats standards tels que Connextra. L'objectif des auteurs est de proposer un dispositif pédagogique qui, au-delà d'une simple correction automatisée, serve de guide pratique pour sensibiliser les utilisateurs aux bonnes pratiques de rédaction.

Sur le plan technique, UST repose sur une combinaison d'heuristiques issues du traitement automatique du langage naturel (NLP) et d'algorithmes d'apprentissage automatique. Le module NLP vérifie la présence des éléments attendus dans le format Connextra (acteur, action, bénéfice) et signale les problèmes de lisibilité, tandis que le module de machine learning fournit une estimation des points de story à partir des caractéristiques linguistiques. En complément, l'outil intègre un module de recommandation basé sur l'API OpenAI, qui suggère des reformulations ou des améliorations contextuelles afin de renforcer la lisibilité et la précision des artefacts produits. Les retours fournis par UST portent notamment sur la conformité au format Connextra, la lisibilité linguistique et l'effort estimé.

L'évaluation empirique de l'outil a impliqué 40 professionnels issus d'équipes agiles, mobilisés pour tester la plateforme et répondre à des questionnaires basés sur deux cadres théoriques largement utilisés en interaction homme-machine : le *Technology Acceptance Model* (TAM) et l'AttrakDiff. Les résultats indiquent une perception globalement positive de l'utilité (*perceived usefulness*) et de l'utilisabilité (*perceived ease of use*) de l'outil. Les participants ont particulièrement apprécié la capacité de l'outil à offrir un retour immédiat sur la structuration et la clarté des user stories, tout en soulignant des limites concernant la contextualisation fine dans des environnements complexes et une certaine dépendance aux

suggestions automatiques.

En termes de limites, il est important de noter que l'évaluation d'UST ne repose sur aucun cadre agile formel d'analyse de la qualité des user stories, tels que les cadres INVEST ou AQUSA. L'outil ne vise pas une conformité rigoureuse aux standards de qualité, mais plutôt une amélioration progressive à visée pédagogique. De plus, l'étude reste exploratoire, avec une taille d'échantillon modeste et un contexte d'expérimentation contrôlé, ce qui limite la généralisation des résultats à de larges environnements industriels.

Par contraste, notre approche dépasse la simple assistance à la rédaction en intégrant explicitement un cadre d'évaluation systématique fondé sur AQUSA. Là où UST met l'accent sur l'accompagnement et la sensibilisation des praticiens, notre méthodologie vise à assurer la conformité stricte des user stories générées aux critères de qualité agile. En particulier, nous ciblons des documents de conception riches et narratifs comme les Game Design Document (GDD), qui posent des défis spécifiques en matière de variabilité et d'interprétation, et pour lesquels un mécanisme d'assurance qualité automatisée s'avère indispensable.

Les travaux examinés dans cette seconde catégorie explorent le potentiel des grands modèles de langage (LLM) pour soutenir l'évaluation et l'amélioration de user stories déjà rédigées. Ces recherches traduisent un intérêt croissant de la communauté scientifique pour l'intégration de l'intelligence artificielle dans les pratiques agiles, à travers différentes approches. Parmi celles-ci, on retrouve l'usage d'agents collaboratifs simulant des rôles Scrum Tiedemann *et al.* (2024), des comparaisons empiriques entre user stories rédigées par des humains et celles générées par des LLM dos Santos *et al.* (2025), ou encore des outils interactifs fournissant un retour en temps réel sur la qualité des stories da Silva Neo *et al.* (2024).

Pris dans leur ensemble, ces travaux permettent de mieux comprendre les défauts fréquemment rencontrés dans les user stories, de proposer des reformulations pertinentes, et de développer des recommandations qualitatives ou semi-automatisées visant à améliorer leur qualité. Toutefois, plusieurs limites méthodologiques se dégagent.

D'abord, aucun de ces travaux ne s'applique au domaine spécifique du développement de jeux vidéo. Les jeux de données utilisés pour l'évaluation proviennent essentiellement de projets logiciels d'entreprise, d'études de cas éducatives ou d'expériences contrôlées, sans prise en compte de documents riches en narration ou en éléments visuels comme les Game Design Document (GDD). Ensuite, bien que certains travaux utilisent des cadres d'évaluation comme QUS, aucun n'intègre une démarche d'assurance qualité fondée sur un cadre formel tel qu'AQUSA. Les évaluations sont généralement qualitatives, basées sur les retours de praticiens, sans que les critères agiles soient ancrés directement dans le processus d'amélioration.

Par contraste, notre recherche vise un domaine d'application encore peu exploré : la génération automatisée de user stories à partir de GDD, documents à forte composante narrative utilisés dans l'industrie vidéoludique. Nous proposons une méthodologie fondée sur la génération par prompts, spécifiquement conçue pour répondre aux particularités des GDD. Celle-ci mobilise trois stratégies distinctes : *Chain-of-Thought*, *Refine-and-Thought* et une approche hybride combinant les deux, afin d'améliorer la cohérence, la structuration et la granularité des stories générées.

Surtout, notre méthode intègre de manière explicite le cadre AQUSA dans la boucle de génération, ce qui permet la détection systématique des défauts et l'amélioration guidée par des critères de qualité agile. Ce positionnement méthodologique fait de notre travail une contribution originale en matière d'automatisation fiable de la documentation agile dans des contextes créatifs, complexes et non structurés comme celui du développement de jeux vidéo.

# CHAPITRE III
# UNE NOUVELLE APPROCHE POUR LA GÉNÉRATION AUTOMATISÉE DE USER STORIES À PARTIR DE GAME DESIGN DOCUMENT

## 3.1 VUE D'ENSEMBLE

La méthodologie proposée dans ce mémoire vise à automatiser la génération et l'amélioration de *user stories* à partir de *Game Design Document* (GDD) en s'appuyant sur les capacités récentes des modèles de langage à grande échelle (LLM). Une telle automatisation répond à plusieurs enjeux : accélérer les phases initiales de développement, renforcer la traçabilité des exigences fonctionnelles et promouvoir une documentation plus cohérente et conforme aux standards agiles.

La démarche repose sur un pipeline modulaire et itératif, conçu pour combiner plusieurs stratégies de génération, intégrer des mécanismes d'évaluation formalisés et permettre une boucle d'amélioration continue. La figure 3.1 illustre cette architecture globale, qui constitue la colonne vertébrale de l'approche expérimentale.

Chaque GDD constitue le point d'entrée du processus et traverse successivement plusieurs modules : (1) un module d'exécution des prompts, qui applique différentes stratégies de génération (CoT, RaT, hybride) à l'aide de plusieurs LLM ; (2) un module d'évaluation, qui diagnostique les défauts de qualité des *user stories* générées à l'aide du cadre AQUSA ; (3) un module de raffinement, qui intègre explicitement ces critères de qualité dans de nouveaux prompts ; et (4) un module de sortie, qui conserve l'ensemble des générations initiales et raffinées pour comparaison.

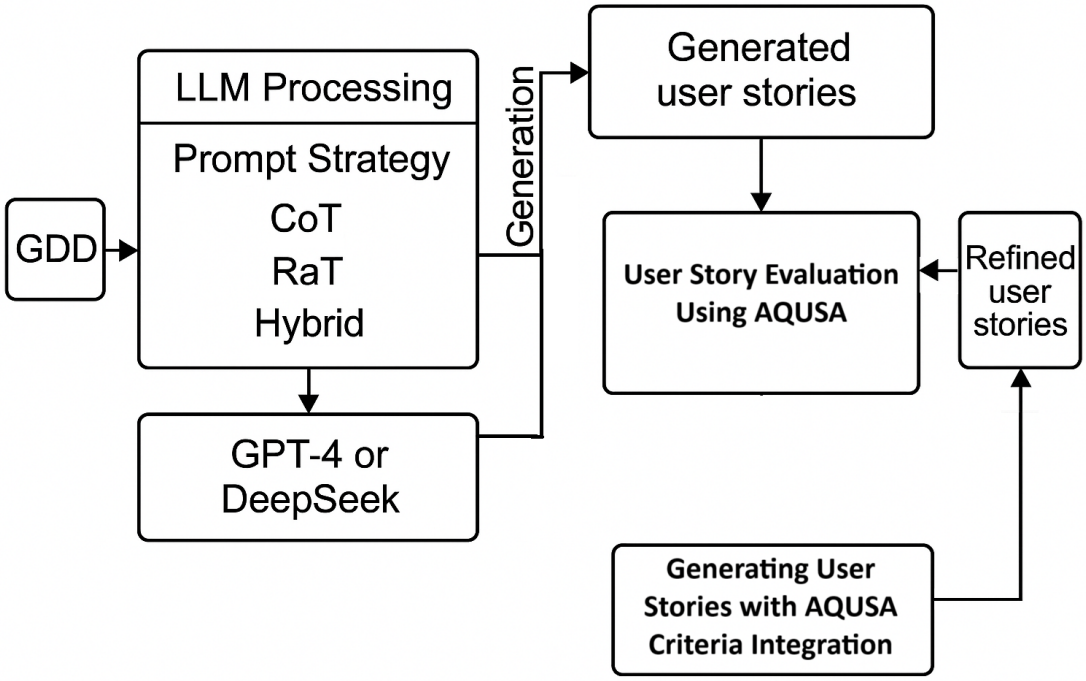


**FIGURE 3.1 : Architecture générale du pipeline de génération et d'évaluation.**

Cette conception repose sur deux principes directeurs. Premièrement, l'hétérogénéité des GDD (industriels ou académiques, narratifs ou techniques) impose une méthodologie robuste, capable de s'adapter à des structures textuelles très diverses. Deuxièmement, la qualité des artefacts générés ne peut être garantie que par l'intégration d'un mécanisme formel de contrôle, tel qu'AQUSA, au sein même de la boucle de génération. En plaçant l'évaluation au cœur du pipeline, cette approche dépasse une simple expérimentation exploratoire pour constituer un véritable dispositif d'assurance qualité.

Enfin, la méthodologie a été conçue dans une optique de généricité et de reproductibilité. Le pipeline proposé est extensible : il peut accueillir de nouveaux modèles de langage, des stratégies de *prompting* alternatives ou d'autres cadres d'évaluation. De plus, tous les choix méthodologiques (corpus, prompts, paramètres et critères d'évaluation) sont documentés en

détail afin de permettre la réplication des expériences et de renforcer la validité scientifique des résultats. Les sections suivantes détaillent successivement le corpus utilisé, les stratégies de *prompting*, le fonctionnement du pipeline et le protocole d'évaluation.

## 3.2 CORPUS ET PRÉPARATION DES DONNÉES

L'étude expérimentale s'appuie sur un corpus de huit *Game Design Document* (GDD) sélectionnés afin de représenter la diversité des pratiques en conception vidéoludique. Quatre d'entre eux proviennent de projets industriels, publiés officiellement ou diffusés par des studios indépendants, tandis que les quatre autres sont issus de contextes académiques, généralement produits dans le cadre de travaux universitaires ou de formations spécialisées.

Cette sélection vise à couvrir une variété de genres vidéoludiques (jeux de rôle, de plate-forme, de stratégie, etc.), ainsi qu'une hétérogénéité en termes de structure, de style narratif et de niveau de détail technique. Les GDD ont été retenus selon deux critères principaux : (1) leur accessibilité publique et (2) la présence d'informations suffisamment riches pour permettre une transformation en artefacts agiles.

Avant leur intégration dans le pipeline de génération, les Game Design Documents (GDD) ont fait l'objet d'un prétraitement indispensable afin d'assurer une extraction stable et exploitable par les modèles. Ce prétraitement comportait trois opérations principales : (1) le nettoyage des sections non pertinentes, (2) la segmentation du contenu en unités textuelles exploitables et (3) la normalisation linguistique.

Dans la pratique, ces opérations n'ont pas pu être entièrement automatisées en raison de l'hétérogénéité des formats, de la qualité variable des fichiers et de l'absence de standardisation structurelle des GDD. Lorsque les documents étaient fournis sous forme de PDF lisibles

(qualité correcte, structure stable), les étapes de nettoyage, de segmentation et de normalisation pouvaient être réalisées de manière largement automatique à l'aide d'outils d'OCR et de scripts simples. Ces cas représentaient une partie du corpus et impliquaient un effort limité.

En revanche, pour les GDD présentant une mise en page dégradée, une faible qualité d'image ou des éléments textuels difficilement reconnaissables par les outils d'OCR, un traitement manuel a été nécessaire. Ce travail incluait l'extraction manuelle de sections, la correction ou la réécriture de passages illisibles, ou encore l'utilisation successive de différents moteurs d'OCR. Ces interventions se sont révélées plus lourdes et chronophages, mais elles étaient essentielles pour garantir la complétude et la cohérence des données en entrée du pipeline.

Ainsi, le prétraitement des GDD s'appuie sur une approche hybride : automatisée lorsque la qualité des documents le permet, et manuelle lorsque la structure ou la lisibilité du fichier impose une intervention humaine. Ce choix méthodologique permet d'assurer une qualité de données suffisante pour les étapes de génération et d'évaluation, tout en tenant compte des contraintes propres aux documents narratifs non standardisés. Les caractéristiques détaillées des GDD et leur niveau de prétraitement seront décrits dans le Chapitre 4.

## 3.3 MODÈLES ET STRATÉGIES DE GÉNÉRATION

La méthodologie s'appuie sur l'utilisation de modèles de langage à grande échelle (*Large Language Models*, LLM), appliqués à des *Game Design Documents* (GDD) afin de produire automatiquement des *user stories*. Deux moteurs distincts ont été sélectionnés : **ChatGPT (GPT-4)** et **DeepSeek**. Ce choix repose sur leur disponibilité, leurs performances reconnues dans des tâches de génération textuelle complexes, ainsi que sur leur complémentarité en matière d'architecture et de conception. L'emploi de deux LLM permet d'examiner les variations

de comportement et de robustesse, et d'évaluer dans quelle mesure les résultats obtenus sont généralisables indépendamment du moteur utilisé.

Les stratégies de génération s'inscrivent directement dans le pipeline présenté au Chapitre 3. Trois routines d'ingénierie du prompt (*prompt engineering*) ont été explorées, chacune correspondant à une logique distincte de raisonnement et d'amélioration Wei *et al.* (2022) :

— **Chain-of-Thought (CoT)** : cette stratégie repose sur un raisonnement explicite formulé étape par étape, afin de guider le modèle dans la décomposition progressive des éléments d'un GDD. L'objectif est de structurer la pensée du modèle et de rendre transparent le passage d'une fonctionnalité identifiée à la *user story* correspondante. Cette approche, fondée sur le raisonnement déductif, a démontré son efficacité pour améliorer la cohérence logique des textes générés dans des tâches complexes nécessitant une segmentation progressive du raisonnement Wei *et al.* (2022).

— **Refine-and-Thought (RaT)** : cette approche introduit une boucle de raffinement. Une première version de la *user story* est générée, puis le modèle produit une justification et une version améliorée. Ce mécanisme vise à corriger les imprécisions initiales et à renforcer la qualité des sorties en intégrant un processus itératif d'amélioration. RaT illustre un paradigme réflexif dans lequel le modèle évalue et améliore ses propres productions, conformément aux travaux récents sur les *self-improving prompts*.

— **Approche Hybride** : cette méthode combine les deux précédentes. Elle s'appuie à la fois sur la structuration pas à pas du raisonnement (CoT) et sur le raffinement successif des sorties (RaT). L'objectif est de capitaliser sur les avantages de chacune des stratégies pour générer des *user stories* à la fois claires, complètes et conformes aux standards Agiles. Cette combinaison constitue une forme de *multi-phase prompting*, dans laquelle

la cohérence du raisonnement est assurée en amont et la qualité finale consolidée par une boucle de correction.

Ces trois stratégies ont été retenues afin de représenter des paradigmes complémentaires du *prompt engineering* : la déduction explicite (CoT), la révision itérative (RaT) et la combinaison raisonnée des deux (Hybride). Elles permettent d'évaluer, à travers un protocole contrôlé, dans quelle mesure la structuration du raisonnement et le raffinement itératif influencent la qualité et la stabilité des artefacts générés.

En appliquant ces trois stratégies avec les deux modèles retenus, chaque GDD a donné lieu à six séries indépendantes de *user stories* (trois stratégies × deux modèles). Cette configuration expérimentale permet non seulement de comparer les résultats selon les approches de génération, mais aussi de mettre en évidence les sensibilités propres à chaque moteur de langage. Ainsi, l'étude ne se limite pas à une comparaison entre prompts, mais explore également l'interaction entre la stratégie adoptée et les caractéristiques intrinsèques du modèle utilisé.

### 3.3.1 EXEMPLES DE PROMPTS

Afin d'assurer la reproductibilité de l'expérience et de permettre une compréhension fine des stratégies appliquées, nous présentons dans cette sous-section les formulations de prompts utilisées pour guider les LLM. Chaque stratégie de génération est illustrée par un extrait représentatif, appliqué de manière uniforme à l'ensemble des GDD.

**Prompt pour la stratégie Chain-of-Thought (CoT).** Ce prompt incite le modèle à expliciter son raisonnement étape par étape avant de formuler la *user story*. L'objectif est de rendre

transparent le cheminement logique qui relie la fonctionnalité identifiée dans le GDD à la user story finale.

```
Context:
You are an expert in agile development and well-versed in the "Chain-of-Thought" method. You are given a Game Design Document (GDD) from which you must extract clear and structured user stories.

Goal:
- Identify all features or gameplay elements described in the GDD.
- Briefly describe the purpose or function of each feature in the game.
- Convert each feature into a user story, using the format:
  "As a [type of user], I want [feature] so that [goal]."

Instructions:
- Carefully read the GDD.
- Explain your reasoning step by step using the "Chain-of-Thought" method.
- Clearly organize your response to show how you move from feature identification to user story formulation.

Expected Format:
- Feature Identification: list all features.
- Purpose of Each Feature: describe briefly why each feature exists.
- User Stories: one story per feature using the specified format.
- Chain-of-Thought Reasoning: make explicit the logic behind your transformation process.

Guidelines:
- Do not skip minor features even if they seem secondary.
- Do not generate all at once. Start with the first 10 features only, then wait for a follow-up instruction.
```

```
- Structure your response for clarity and reusability by a development team.
```

Listing 3.1 – Initial prompt for CoT strategy applied to all GDD

**Prompt pour la stratégie Refine-and-Thought (RaT).** Ce prompt met en place une boucle de raffinement, dans laquelle le modèle génère une première version de la *user story*, produit une justification, puis propose une version améliorée. Cette approche vise à corriger les imprécisions initiales et à enrichir la qualité des stories générées.

```
You are an agile expert applying the "Refine-and-Thought" (RaT) method to transform a Game Design Document (GDD) into high-quality user stories.

Rules (very important):
- Work strictly from the provided GDD. Do NOT invent features that are not present.
- Enforce the Connextra form exactly: "As a <role>, I want <action> so that <benefit>."
- Enforce atomicity: one behavior per story. If a sentence contains multiple actions, split into multiple stories.
- Avoid vague terms: do not use etc., maybe, possibly, some, various.
- Produce only the first 10 features, then stop and wait for "continue".
- Use the output schema below for every feature.

Quality criteria (AQUSA checklist to fill for each story: Yes/No):
Role present; Action present; Benefit present; Atomic (one action); Minimal (no implementation details); Uniform (Connextra); Testable; Valuable.

For each detected feature, follow the 3 RaT steps and the output schema:

1) Initial Extraction
```

```
   - Draft the initial user story (Connextra).
2) Refinement
   - Improve clarity and atomicity (split if needed); add missing role/benefit; keep it minimal and testable.
3) Reflection
   - Briefly justify changes (1–2 sentences).
   - Add up to 3 acceptance criteria in Given/When/Then form.

Traceability
   - Cite the exact GDD anchor used (e.g., section heading + a short quote of ≤ 20 words).

OUTPUT SCHEMA (repeat for each feature; this exact Markdown structure is required):

### Feature <ID>: <short name>
GDD Anchor: <section/heading>; Quote: "≤ 20 words taken from GDD>"

Initial Story:
"As a <role>, I want <action> so that <benefit>."

Refined Story:
"As a <role>, I want <action> so that <benefit>."

Acceptance Criteria (max 3):
- Given <context>, When <event>, Then <outcome>.
- Given <context>, When <event>, Then <outcome>.
- (optional)

Reflection:
```

```
<1  2  sentences explaining the improvements and why the refined story is better for the team.>

AQUSA Checklist:
- Role: Yes/No; Action: Yes/No; Benefit: Yes/No; Atomic: Yes/No; Minimal: Yes/No; Uniform: Yes/No; Testable: Yes/No; Valuable: Yes/No.

Now process the provided GDD and output only the first 10 features using the schema above. Then stop.
```

Listing 3.2 – Prompt used for Refine-and-Thought (RaT) strategy

**Prompt pour la stratégie Hybride.** Cette formulation combine les deux mécanismes précédents. Le prompt guide le modèle à raisonner étape par étape (CoT), puis à raffiner la sortie de manière itérative (RaT). L'objectif est de tirer parti de la complémentarité entre structuration séquentielle et amélioration progressive.

```
You are an agile expert combining Chain-of-Thought (CoT) with Refine-and-Thought (RaT) to extract and improve user stories from a Game Design Document (GDD).

Rules (very important):
- Work strictly from the provided GDD. Do NOT invent features that are not present.
- Enforce Connextra exactly: "As a <role>, I want <action> so that <benefit>."
- Enforce atomicity: one behavior per story. If multiple actions exist, split into multiple stories.
- Keep stories minimal (no UI/implementation details unless essential for value/testability).
- Avoid vague terms: do not use etc., maybe, possibly, some, various.
```

```
- Produce only the first 10 features, then stop and wait for "continue".
- Use the output schema below for every feature.

Quality criteria:
- AQUSA checklist (fill Yes/No for each story): Role, Action, Benefit, Atomic, Minimal, Uniform (Connextra), Testable, Valuable.
- (Optional) INVEST snapshot (Y/N): Independent, Negotiable, Valuable, Estimable, Small, Testable.

Process for each detected feature:
1) CoT Feature Identification
   - Identify the feature and briefly show your reasoning (2  4   bullet points).
2) Initial Story (Connextra)
   - Draft the initial user story.
3) RaT Refinement
   - Improve clarity/atomicity; add missing role or benefit; keep minimal; ensure testability.
4) Reflection
   - Justify the refinements in 1  2   sentences (why the refined version is better).
5) Acceptance Criteria
   - Add up to 3 criteria in Given/When/Then form.

Traceability:
- Cite the exact GDD anchor used (e.g., section heading + a short quote of  20   words).

OUTPUT SCHEMA (repeat for each feature; this exact Markdown structure is required):

### Feature <ID>: <short name>
```

```
GDD Anchor: <section/heading>; Quote: "< 20 words from GDD>"

CoT Reasoning:
- <bullet 1>
- <bullet 2>
- (optional) <bullet 3 4 >

Initial Story:
"As a <role>, I want <action> so that <benefit>."

Refined Story:
"As a <role>, I want <action> so that <benefit>."

Acceptance Criteria (max 3):
- Given <context>, When <event>, Then <outcome>.
- Given <context>, When <event>, Then <outcome>.
- (optional)

Reflection:
<1 2 sentences explaining the improvements and their impact on user value and development.>

AQUSA Checklist:
- Role: Y/N; Action: Y/N; Benefit: Y/N; Atomic: Y/N; Minimal: Y/N; Uniform: Y/N; Testable: Y/N; Valuable: Y/N.

INVEST (optional):
- I: Y/N; N: Y/N; V: Y/N; E: Y/N; S: Y/N; T: Y/N.

Now process the provided GDD and output only the first 10 features.
```

Listing 3.3 – Prompt used for Hybrid strategy (Chain-of-Thought + Refine-and-Thought)

### 3.3.2 INTÉGRATION DES CRITÈRES AQUSA DANS LES PROMPTS

Une spécificité centrale de notre méthodologie réside dans l'intégration explicite des critères de qualité issus du cadre *agile Quality User Story Assessment* (AQUSA) dans la formulation des *prompts*. Cette approche vise à orienter les modèles de langage vers la production de *user stories* conformes, dès leur conception, aux standards de qualité reconnus dans la littérature agile.

Les dimensions de qualité définies par AQUSA, telles que l'atomicité, la clarté, la valeur utilisateur, la testabilité et l'uniformité ont été reformulées sous forme de consignes explicites directement intégrées aux *prompts*. Plutôt que de se limiter à demander la transformation de fonctionnalités issues des GDD, les instructions exigent que chaque *user story* générée respecte ces principes fondamentaux. Par exemple, une directive type rappelle qu'une *user story* doit inclure un acteur explicite, exprimer une action observable, mentionner un bénéfice mesurable et demeurer suffisamment concise pour être implémentée dans un seul incrément agile.

Cette intégration poursuit un double objectif. D'une part, elle réduit la probabilité de défauts fréquemment détectés par AQUSA, tels que l'absence d'acteur, les formulations vagues ou les bénéfices mal définis. D'autre part, elle renforce la boucle d'amélioration continue décrite à la Figure 3.1 : en produisant dès le départ des artefacts mieux alignés sur les standards agiles, le module de raffinement ultérieur s'appuie sur une base de génération plus stable et de meilleure qualité.

```
Guidelines:
- Ensure each user story is atomic (only one functionality).
- Follow the Connextra format: "As a [actor], I want [feature] so that ↩
    ↪[benefit]."
- Avoid ambiguity: the story must be clear and concise.
- Explicitly highlight the user value in each story.
- Make the story testable by providing a measurable outcome.
```

Listing 3.4 – Extrait simplifié illustrant l'intégration des critères AQUSA dans un prompt

Cette adaptation des *prompts* constitue une étape méthodologique clé qui distingue notre approche des travaux existants. Contrairement aux études qui évaluent la qualité uniquement *a posteriori*, notre pipeline favorise une conformité proactive aux standards de qualité, tout en maintenant une boucle d'évaluation et de raffinement pour éliminer les défauts résiduels.

## 3.4 PIPELINE DE GÉNÉRATION ET DE RAFFINEMENT

L'architecture méthodologique proposée repose sur un pipeline modulaire conçu pour automatiser et améliorer la génération de *user stories* à partir de *Game Design Document* (GDD). Cette structure vise à garantir la traçabilité, la flexibilité et la reproductibilité du processus, tout en intégrant une boucle complète de génération, d'évaluation et de raffinement. Le pipeline est représenté à la Figure 3.1 et formalisé dans l'Algorithme 3.1.

### 3.4.1 ARCHITECTURE MODULAIRE

Le pipeline est composé de cinq modules complémentaires :

— **Module d'entrée.** Il prend en charge les GDD fournis sous forme textuelle, qu'ils

soient issus de projets académiques ou industriels. Un prétraitement minimal (nettoyage et uniformisation) est appliqué afin de préserver la richesse narrative du document et d'évaluer la robustesse des modèles face à des données réelles et hétérogènes.

— **Module d'exécution des prompts.** Chaque GDD est traité selon trois stratégies de génération (CoT, RaT, Hybride) et deux modèles (ChatGPT et DeepSeek), aboutissant à six combinaisons expérimentales. Ce module orchestre l'envoi des prompts, la collecte des sorties et leur organisation en jeux de résultats intermédiaires.

— **Module d'évaluation.** Les *user stories* générées sont automatiquement analysées par l'outil *AQUSA*, qui détecte les défauts liés à l'atomicité, la clarté, la valeur utilisateur et la testabilité. Les résultats sont sauvegardés sous forme de rapports structurés, servant d'entrée à la phase de raffinement.

— **Module de raffinement.** À partir des diagnostics AQUSA, de nouveaux prompts sont élaborés pour corriger les défauts identifiés. Cette boucle d'amélioration permet d'obtenir une seconde version des *user stories* mieux alignée sur les critères de qualité agile, sans intervention humaine directe.

— **Module de sortie.** Il centralise l'ensemble des résultats : versions initiales et raffinées, évaluations AQUSA et métadonnées expérimentales afin d'assurer la traçabilité et de permettre une comparaison systématique entre stratégies, modèles et itérations.

### 3.4.2 SÉQUENCE DE TRAITEMENT

Le pipeline suit cinq étapes : (1) extraction des fonctionnalités à partir du GDD ; (2) génération initiale des *user stories* selon la stratégie de *prompting* ; (3) évaluation automatique par AQUSA pour détecter les défauts de qualité ; (4) régénération et raffinement guidés par les résultats de cette évaluation ; et (5) archivage et comparaison des versions produites.

Ces étapes correspondent directement aux modules décrits ci-dessus et sont illustrées dans la Figure 3.1. Leur enchaînement méthodologique garantit la reproductibilité du processus et la cohérence des analyses comparatives présentées au Chapitre 4.

### 3.4.3 PSEUDO-CODE DU PIPELINE

**Entrées :** Corpus de GDD $\mathscr{D}$

**Output :** Pour chaque $(d, m, s)$ : ensembles $\{us_{init}, eval_{init}, us_{ref}, eval_{ref}\}$

**pour chaque** *GDD* $d \in \mathscr{D}$ **faire**
  Prétraiter $d$ (nettoyage minimal);
  **pour chaque** *modèle* $m \in \{ChatGPT, DeepSeek\}$ **faire**
    **pour chaque** *stratégie* $s \in \{CoT, RaT, Hybride\}$ **faire**
      Générer les *user stories* initiales $us_{init}$ à partir de $d$ avec $(s, m)$;
      Évaluer $us_{init}$ avec AQUSA $\rightarrow eval_{init}$;
      Construire un prompt raffiné $p_{ref}$ basé sur $eval_{init}$ et les critères AQUSA;
      Générer les *user stories* raffinées $us_{ref}$ à partir de $d$ avec $(s, m, p_{ref})$;
      Évaluer $us_{ref}$ avec AQUSA $\rightarrow eval_{ref}$;
      Sauvegarder $\{us_{init}, eval_{init}, us_{ref}, eval_{ref}\}$;
    **fin**
  **fin**
**fin**

*Chaque GDD donne lieu à six combinaisons expérimentales : trois stratégies de prompting appliquées à deux modèles de langage.*

**Algorithme 3.1 : Pipeline de génération et de raffinement des *user stories***

## 3.5 ÉVALUATION DE LA QUALITÉ

### 3.5.1 AQUSA

L'évaluation de la qualité des *user stories* générées s'appuie sur le cadre d'évaluation automatisé *agile Quality User Story Analyzer* (AQUSA), conçu pour diagnostiquer les défauts récurrents dans les artefacts agiles rédigés en langage naturel Ferrari *et al.* (2017). Ce module intervient au cœur du pipeline présenté précédemment, assurant un contrôle objectif et reproductible de la qualité des productions issues des modèles de langage. AQUSA a été retenu car il fournit une analyse systématique, fondée sur des critères issus des bonnes pratiques agiles, et constitue ainsi un complément méthodologique essentiel à l'utilisation des LLM dans la génération de *user stories*.

Plus précisément, AQUSA évalue chaque *user story* selon plusieurs dimensions de qualité :

— **Atomicité** : la story doit correspondre à une seule fonctionnalité indivisible ;

— **Clarté** : la formulation doit être compréhensible par toutes les parties prenantes sans ambiguïté ;

— **Uniformité** : l'ensemble des stories doit adopter une structure homogène, notamment le format Connextra (*« En tant que [acteur], je veux [fonctionnalité] afin de [bénéfice] »*) ;

— **Valeur utilisateur** : la story doit expliciter le bénéfice concret apporté à l'acteur ou à l'utilisateur cible ;

— **Testabilité** : la fonctionnalité décrite doit être vérifiable au moyen d'un scénario de test objectif ;

— **Minimalisme** : la story doit éviter tout élément superflu ou non actionnable.

L'outil repose sur une combinaison de règles linguistiques et de modèles syntaxiques

destinés à détecter automatiquement les défauts associés à ces critères. Par exemple, l'absence d'un acteur explicite indique souvent un défaut de rôle, tandis qu'un bénéfice exprimé de manière vague (ex. : *« être satisfait »*) est signalé comme imprécis. Chaque anomalie détectée est annotée dans un rapport textuel, associant chaque *user story* au type de défaut identifié.

Ces rapports constituent l'entrée du module de raffinement décrit à la Section 3.4, permettant d'intégrer les diagnostics AQUSA dans les prompts de seconde génération afin d'améliorer itérativement la qualité des artefacts produits.

### 3.5.2 MESURES QUANTITATIVES

Les sorties produites par AQUSA permettent de dériver plusieurs indicateurs quantitatifs caractérisant la qualité des *user stories*. Dans la présente étude, quatre mesures principales ont été retenues :

- le **nombre moyen de défauts par *user story***, indicateur de la densité de problèmes détectés par l'analyse automatique ;
- la **distribution des types de défauts**, permettant d'identifier les erreurs les plus fréquentes (par exemple, absence d'acteur, ambiguïté ou non-testabilité) ;
- le **taux de conformité**, calculé comme le pourcentage de *user stories* ne présentant aucun défaut détecté par AQUSA, représentant ainsi un indicateur synthétique de qualité globale ;
- les **améliorations relatives**, définies comme la variation du nombre moyen de défauts entre la génération initiale et la génération raffinée, afin de mesurer l'efficacité du processus de raffinement.

Ces mesures sont calculées pour chaque combinaison de modèle (ChatGPT, DeepSeek)

et de stratégie de *prompting* (CoT, RaT, Hybride), ce qui permet de comparer objectivement leurs performances respectives. L'agrégation des résultats sur l'ensemble des GDD fournit ensuite une vue d'ensemble des tendances et sert de base aux analyses présentées dans le Chapitre 4 consacré aux résultats expérimentaux.

## 3.6 SYNTHÈSE

Ce chapitre a présenté la méthodologie élaborée pour automatiser la génération et l'amélioration de *user stories* à partir de *Game Design Document* (GDD) en exploitant les capacités des modèles de langage à grande échelle (LLM). L'approche repose sur un pipeline structuré et modulaire, intégrant plusieurs stratégies de *prompting* (Chain-of-Thought, Refine-and-Thought et approche hybride) appliquées à deux modèles distincts (ChatGPT et DeepSeek). Ce dispositif expérimental permet de produire, pour chaque GDD, plusieurs ensembles de *user stories* exploitables, favorisant ainsi une comparaison systématique entre stratégies et moteurs de génération.

La méthodologie intègre également une boucle d'évaluation et de raffinement fondée sur l'outil *AQUSA*, garantissant une mesure objective de la qualité selon des critères agiles tels que la clarté, l'atomicité, la testabilité et la cohérence. Cette étape vise à identifier les défauts récurrents dans les générations initiales et à alimenter un processus de régénération guidé par des prompts enrichis, afin d'accroître la conformité des *user stories* aux standards agiles.

L'ensemble des choix méthodologiques répond aux objectifs et aux questions de recherche formulés dans l'introduction. D'une part, l'utilisation conjointe de GDD industriels et académiques permet de tester la robustesse de l'approche sur des documents variés en termes de style, de structure et de complexité. D'autre part, la comparaison entre stratégies de *prompting* et modèles de langage offre une compréhension fine des facteurs influençant la qualité des

artefacts générés. Enfin, l'intégration explicite d'un cadre d'évaluation systématique confère à l'étude un haut degré de rigueur et de reproductibilité.

Ainsi, la méthodologie proposée établit une base solide pour l'analyse expérimentale. Elle s'aligne directement sur les objectifs formulés dans l'introduction, en apportant des réponses concrètes aux questions de recherche suivantes : (1) dans quelle mesure les modèles de langage (LLM) peuvent-ils générer automatiquement des *user stories* cohérentes à partir de Game Design Document ? (2) comment l'intégration de critères de qualité issus d'AQUSA dans les *prompts* influence-t-elle la conformité des artefacts produits ? et (3) quelle stratégie de *prompting* offre le meilleur équilibre entre exhaustivité, clarté et testabilité ?

L'ensemble du dispositif méthodologique combinant extraction, génération, évaluation et raffinement a été conçu pour répondre à ces interrogations de manière systématique et reproductible.

Le chapitre suivant présentera les résultats expérimentaux issus de cette approche, en comparant les performances des différents modèles et stratégies selon les critères de qualité définis.

# CHAPITRE IV

# RÉSULTATS

## 4.1 PRÉSENTATION SYNTHÉTIQUE DU CORPUS ÉVALUÉ

Le corpus utilisé pour l'évaluation expérimentale comprend huit *Game Design Document* (GDD), répartis en deux catégories : documents commerciaux et documents académiques. Contrairement au Chapitre 3, qui détaillait les critères de sélection, la présente section vise à synthétiser les caractéristiques des documents effectivement exploités afin de situer les résultats dans leur contexte de production.

Cette double typologie du corpus n'a pas seulement une valeur descriptive : elle permet également d'évaluer la capacité des modèles de langage à généraliser entre des contextes de conception distincts d'une part, des GDD commerciaux riches et formalisés, et d'autre part, des GDD académiques plus concis et hétérogènes. Cette diversité constitue donc un levier méthodologique essentiel pour tester la robustesse et la généricité de l'approche proposée.

### 4.1.1 GDD COMMERCIAUX

Quatre GDD proviennent de projets de jeux vidéo emblématiques, rendus publics par leurs auteurs sous forme de documents historiques ou de présentations conceptuelles.

*Grand Theft Auto* (12 pages) [1] constitue l'un des premiers pitchs du jeu, décrivant les fondements d'un monde ouvert centré sur des missions criminelles, sans être accompagné d'artefacts agile.

1. http://gamedevs.org/uploads/grand-theft-auto.pdf

*Diablo* (8 pages) [2] est un document de type « pitch » exposant la vision initiale d'un RPG hack'n'slash, mettant l'accent sur l'atmosphère et les mécaniques fondamentales.

*BioShock* (17 pages) [3] présente une conception narrative et systémique riche, centrée sur un univers dystopique et des choix moraux du joueur.

Enfin, le *Doom Bible* (79 pages) [4] constitue un exemple particulièrement volumineux, détaillant les mécaniques, les niveaux et les personnages d'un futur FPS culte.

Ces documents se distinguent par leur richesse conceptuelle et leur variété stylistique, mais aucun ne contient de user stories, ce qui en fait un terrain idéal pour tester la génération automatique.

### 4.1.2 GDD ACADÉMIQUES

Les quatre autres documents proviennent de projets réalisés dans un cadre universitaire, dans le contexte de cours en conception de jeux.

*Lost in Dino World* (4 pages) décrit un jeu de plateforme 2D pour mobile, où un scientifique voyage accidentellement à l'époque des dinosaures dans un univers de science-fiction.

*Car Crash BOOM!!! Yeah!* (4 pages) propose un jeu de course coopératif de style cartoon, combinant pilotage et tir dans un cadre compétitif.

---

2. http://www.graybeardgames.com/download/diablo_pitch.pdf
3. https://www.systemshock.org/index.php?topic=2121.msg21031
4. http://5years.doomworld.com/doombible/doombible.pdf

*Little Red Fighting Hood* (7 pages) transpose un conte de fées dans un univers ludique de type *endless runner*, où le joueur collecte de la nourriture tout en affrontant des loups.

Enfin, *Squeeky* (11 pages) est un jeu de plateforme 3D orienté speedrun, mettant en scène un écureuil volant devant accumuler des ressources pour atteindre un abri.

Ces GDD académiques incluaient parfois des user stories rédigées par les étudiants, mais leur qualité restait limitée, ce qui reflète leur rôle essentiellement pédagogique. Leur inclusion dans le corpus permet d'évaluer l'efficacité de notre approche dans des contextes moins formalisés que ceux des documents commerciaux.

La combinaison de ces deux catégories de GDD commerciaux riches mais dépourvus de *user stories*, académiques plus courts et parfois accompagnés de stories rudimentaires offre une diversité précieuse pour évaluer la robustesse et la généricité de notre approche basée sur les LLM.

Cette diversité de sources garantit une évaluation robuste des capacités des modèles à généraliser entre des contextes narratifs et techniques variés, tout en assurant une représentativité suffisante pour comparer les stratégies de génération dans des environnements contrastés. Elle constitue ainsi une base empirique solide pour les analyses quantitatives présentées dans la section suivante.

Cette description du corpus fournit le cadre empirique nécessaire à l'analyse quantitative des résultats. La section suivante présente les statistiques descriptives issues des générations automatiques, afin d'établir un aperçu global du matériel produit avant son évaluation qualitative.

## 4.2 STATISTIQUES DESCRIPTIVES SUR LES USER STORIES GÉNÉRÉES

Avant d'examiner la qualité des *user stories* selon les critères AQUSA, il est nécessaire de caractériser le matériel produit afin d'en évaluer la cohérence, la variabilité et la reproductibilité. Les statistiques descriptives présentées dans cette section offrent une vision d'ensemble des volumes générés, de la longueur moyenne des stories et de leur distribution selon les modèles et les stratégies de *prompting* utilisés. Elles constituent ainsi une base quantitative essentielle pour interpréter, par la suite, les résultats de l'évaluation de qualité.

### 4.2.1 VOLUMES DE GÉNÉRATION

Au total, huit *Game Design Document* (GDD) ont été traités, chacun donnant lieu à six combinaisons distinctes (3 stratégies de *prompting* × 2 modèles de langage). Pour chaque combinaison, plusieurs dizaines à plusieurs centaines de *user stories* ont été produites, avant l'application d'un échantillonnage aléatoire destiné à l'évaluation AQUSA.

Les valeurs présentées dans le tableau 4.1 proviennent d'intervalles observés lors de plusieurs exécutions indépendantes du pipeline de génération. Chaque combinaison de modèle de langage (ChatGPT, DeepSeek) et de stratégie de *prompting* (CoT, RaT, Hybride) a été exécutée entre 5 et 8 fois selon les GDD et la qualité du prétraitement, constituant ainsi la taille d'échantillon utilisée pour l'estimation des volumes générés. Les intervalles reflètent ainsi l'ordre de grandeur et l'amplitude typique du nombre de *user stories* générées avant échantillonnage, mettant en évidence la variabilité naturelle propre aux comportements non déterministes des LLM.

Afin d'obtenir une représentation synthétique comparable entre GDD et stratégies, chaque intervalle a été converti en une valeur moyenne accompagnée d'une mesure de

dispersion. Cette dispersion correspond à la demi-largeur de l'intervalle observé et ne constitue pas un écart-type empirique issu de valeurs individuelles, mais une estimation descriptive destinée à faciliter l'analyse comparative.

**TABLEAU 4.1 : Volume moyen estimé de *user stories* générées avant échantillonnage (moyenne ± dispersion), sur 5 à 8 exécutions indépendantes selon les GDD.**

| Catégorie | GDD | CoT | RaT | Hybride |
|---|---|---|---|---|
| Commercial | GTA (12 p.) | 150 (±10) | 130 (±10) | 140 (±10) |
| Commercial | Diablo (8 p.) | 120 (±10) | 110 (±10) | 125 (±10) |
| Commercial | BioShock (17 p.) | 170 (±10) | 160 (±10) | 175 (±10) |
| Commercial | Doom Bible (79 p.) | 280 (±20) | 260 (±20) | 290 (±20) |
| Académique | Lost in Dino World (4 p.) | 90 (±10) | 80 (±10) | 95 (±10) |
| Académique | Car Crash BOOM ! ! ! (4 p.) | 85 (±10) | 80 (±10) | 95 (±10) |
| Académique | Little Red Fighting Hood (7 p.) | 100 (±10) | 95 (±10) | 105 (±10) |
| Académique | Squeeky (11 p.) | 130 (±10) | 120 (±10) | 135 (±10) |
| **Total estimé** | | **1 225 (±90)** | **1 035 (±90)** | **1 260 (±90)** |

*Note : La valeur entre parenthèses correspond à une estimation de la dispersion calculée à partir de la demi-largeur des intervalles observés. Elle ne constitue pas un écart-type empirique, mais une approximation destinée à représenter la variabilité typique des volumes générés, calculée sur des tailles d'échantillon variant entre 5 et 8 exécutions indépendantes.*

Les volumes obtenus se révèlent globalement stables d'une exécution à l'autre, et les écarts observés demeurent cohérents avec les variations attendues pour des modèles génératifs non déterministes. Ainsi, les différences analysées ultérieurement entre stratégies de *prompting* reflètent principalement des effets méthodologiques plutôt que des fluctuations aléatoires du processus de génération.

### 4.2.2 LONGUEUR DES *USER STORIES*

La longueur des *user stories* constitue un indicateur pertinent du niveau de détail textuel produit par les modèles et stratégies de *prompting*. Afin de caractériser cette dimension, nous avons mesuré la longueur moyenne (en nombre de mots) ainsi que l'écart-type associé pour chaque combinaison modèle–stratégie, en s'appuyant sur un échantillon global de **user stories issues de deux GDD représentatifs** : un GDD académique court (*Lost in Dino World*) et un GDD commercial volumineux (*BioShock Pitch*).

Ces deux documents ont été sélectionnés pour refléter l'amplitude du corpus complet : d'une part, un GDD synthétique et peu détaillé, et d'autre part, un GDD long, structuré et riche en mécaniques. Les résultats observés pour ces deux cas sont cohérents avec ceux obtenus sur l'ensemble des huit GDD, et permettent ainsi de présenter des valeurs statistiques représentatives du comportement général du pipeline.

Les longueurs ont été calculées sur l'ensemble des *user stories* générées pour chaque combinaison modèle–stratégie. Les valeurs sont reportées sous la forme **moyenne (écart-type)**, et *n* correspond au nombre total de *user stories* analysées pour la combinaison considérée.

**TABLEAU 4.2 : Longueur des *user stories* (en mots) selon les modèles et stratégies — moyenne (écart-type) et taille d'échantillon *n*.**

| Modèle | Stratégie | Longueur moyenne (écart-type) | *n* |
|---|---|---|---|
| GPT-4 | CoT | 20.1 (1.9) | 30 |
| GPT-4 | RaT | 26.3 (4.0) | 49 |
| GPT-4 | Hybride | 31.1 (3.3) | 30 |
| midrule DeepSeek | CoT | 24.0 (2.6) | 30 |
| DeepSeek | RaT | 16.5 (2.4) | 100 |
| DeepSeek | Hybride | 44.7 (4.6) | 15 |

De manière générale, les longueurs observées demeurent relativement stables d'un GDD à l'autre et consistent avec les tendances identifiées pour l'ensemble du corpus. GPT-4 génère des *user stories* comprises entre 20 et 31 mots selon la stratégie, tandis que DeepSeek présente un comportement plus contrasté : la stratégie RaT produit les *user stories* les plus concises (environ 16 mots), alors que la stratégie Hybride génère systématiquement les plus longues (environ 45 mots).

Ces résultats indiquent que les différences observées entre modèles et stratégies tiennent davantage au mode de structuration du contenu qu'au volume textuel produit. La variabilité faible à modérée des longueurs suggère que les analyses qualitatives et quantitatives conduites dans les sections suivantes ne sont pas influencées par des variations importantes de taille, mais reflètent principalement des différences méthodologiques dans la génération.

### 4.2.3 RÉPARTITION PAR TYPE DE GDD

L'analyse de la longueur des *user stories* en fonction du type de GDD met en évidence des tendances structurelles cohérentes entre documents commerciaux et académiques. En moyenne, les GDD commerciaux génèrent des *user stories* plus longues, avec des valeurs typiques situées entre 26 et 32 mots selon les modèles et stratégies. À l'inverse, les GDD académiques produisent des *user stories* plus concises, généralement comprises entre 20 et 25 mots.

Cette différence s'explique par la densité informationnelle distincte des deux familles de GDD. Les documents commerciaux sont plus détaillés, couvrent des mécaniques complexes, et contiennent des descriptions narratives plus riches, ce qui amène les modèles de langage à produire des formulations plus longues et plus structurées. Les GDD académiques, souvent

synthétiques et focalisés sur quelques mécaniques clés, induisent quant à eux des productions plus courtes et plus directes.

Ces résultats suggèrent que les LLM adaptent effectivement la longueur de leurs sorties au niveau de détail présent dans le document source. Cette sensibilité au contenu d'entrée, déjà observée dans des travaux antérieurs sur la génération contrôlée, confirme que la structure du GDD influence autant le style que la densité des *user stories* produites.

## 4.3 ÉVALUATION DES USER STORIES GÉNÉRÉES DANS LA PHASE INITIALE

Conformément au protocole décrit au Chapitre 3, l'évaluation de la qualité des *user stories* générées a été réalisée au moyen du cadre *agile Quality User Story Analyzer* (AQUSA). Ce cadre, détaillé précédemment dans la section consacrée à l'évaluation méthodologique, permet de diagnostiquer automatiquement plusieurs types de défauts de qualité (atomicité, minimalisme, uniformité, clarté, testabilité et valeur utilisateur) à partir de règles linguistiques et heuristiques.

Cette première phase visait à mesurer la capacité intrinsèque des modèles de langage à produire des *user stories* exploitables directement à partir de GDD narratifs, sans contrainte de qualité explicite. L'ensemble des sorties a été échantillonné afin d'obtenir un corpus équilibré pour l'évaluation automatique. Au total, 960 *user stories* ont été analysées (8 GDD × 3 stratégies de *prompting* × 2 modèles × 20 stories). Toutes les évaluations ont été réalisées sur les textes en anglais, afin d'assurer la compatibilité avec les règles heuristiques d'AQUSA. Les défauts détectés ont ensuite été agrégés pour identifier les tendances dominantes par critère et par combinaison modèle–stratégie.

### 4.3.1 DISTRIBUTION DES DÉFAUTS IDENTIFIÉS

Les résultats mettent en évidence trois types de défauts prédominants dans cette phase initiale :

— **Minimalisme** (45 %) un grand nombre de *user stories* comportent des formulations redondantes ou des détails excessifs, réduisant leur lisibilité et leur concision dans un backlog agile ;

— **Atomicité** (35 %) — plusieurs *user stories* regroupent plusieurs objectifs ou fonctionnalités, compromettant leur indépendance et leur testabilité ;

— **Uniformité** (10 %) — certaines stories s'écartent du format standard Connextra, entraînant une hétérogénéité structurelle entre les artefacts générés.

Les autres critères : clarté, valeur utilisateur et testabilité n'ont été affectés que marginalement, ce qui suggère que les modèles produisent des textes syntaxiquement cohérents, mais dont la structure reste éloignée des standards agiles.

**TABLEAU 4.3 : Répartition des défauts détectés par AQUSA dans la phase initiale (valeurs représentatives issues d'exécutions multiples du pipeline).**

| Type de défaut | Proportion observée | Exemple typique |
|---|---|---|
| Minimalisme | 45 % | "As a player, I want to collect coins and explore levels with detailed options and backstory." |
| Atomicité | 35 % | "As a gamer, I want to save progress and unlock characters so that I can continue easily." |
| Uniformité | 10 % | "I want to fight enemies for rewards." (absence d'acteur explicite) |

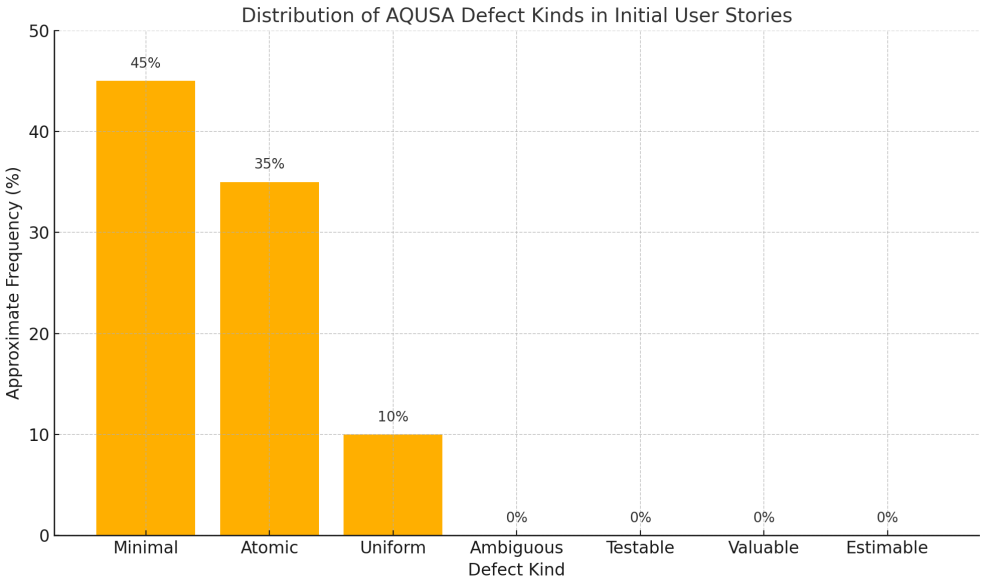


**FIGURE 4.1 : Répartition des défauts identifiés par AQUSA dans les *user stories* générées lors de la phase initiale.**

Ces résultats confirment que, sans contraintes explicites de qualité, les modèles de langage privilégient la cohérence narrative au détriment de la granularité fonctionnelle. Cette tendance se traduit par la génération de stories multi-objectifs et par une faible homogénéité syntaxique. Les omissions de rôles explicites ou de bénéfices utilisateur sont également fréquentes, soulignant les limites de l'autonomie des LLM dans la production d'artefacts agiles. Ces constats ont motivé la mise en œuvre d'une phase de raffinement visant à intégrer les critères AQUSA directement dans les prompts de génération.

### 4.3.2 COMPARAISON ENTRE MODÈLES ET STRATÉGIES

Une analyse comparative a été menée afin d'évaluer l'influence des trois stratégies de *prompting* (CoT, RaT, Hybride) et des deux modèles de langage (ChatGPT et DeepSeek) sur la nature et la fréquence des défauts détectés. Les résultats reposent sur l'évaluation de **960**

***user stories*** issues de la phase initiale de génération, réparties équitablement entre les six combinaisons modèle–stratégie.

Les tendances observées sont les suivantes :

— La stratégie *Chain-of-Thought* (CoT) engendre davantage de défauts de **minimalisme**, en raison de descriptions plus verbeuses et narratives ;

— La stratégie *Refine-and-Thought* (RaT) atténue ces excès mais accroît les défauts d'**atomicité**, les raffinements successifs introduisant parfois plusieurs objectifs dans une même *user story* ;

— La méthode *Hybride* présente des résultats intermédiaires, tout en améliorant l'**uniformité** syntaxique grâce à sa structure mixte ;

— Du point de vue des modèles, **ChatGPT** affiche une meilleure stabilité structurelle et un respect plus constant du format Connextra, tandis que **DeepSeek** génère des formulations plus libres mais légèrement moins homogènes.

**TABLEAU 4.4 : Comparaison des défauts moyens détectés selon les modèles et stratégies (valeurs indicatives issues de l'illustration expérimentale, $n = 960$ *user stories* évaluées).**

| Combinaison | Minimalisme | Atomicité | Uniformité | Total défauts |
|---|---|---|---|---|
| ChatGPT – CoT | 47 % | 33 % | 8 % | 88 % |
| ChatGPT – RaT | 39 % | 41 % | 7 % | 87 % |
| ChatGPT – Hybride | 42 % | 34 % | 9 % | 85 % |
| DeepSeek – CoT | 50 % | 36 % | 12 % | 98 % |
| DeepSeek – RaT | 44 % | 39 % | 10 % | 93 % |
| DeepSeek – Hybride | 45 % | 35 % | 11 % | 90 % |

En moyenne, ChatGPT présente environ **7 à 10 % de défauts en moins** que DeepSeek, principalement sur les critères d'uniformité et de clarté. De même, la stratégie *Hybride* réduit

d'environ **4 à 6 %** le nombre total de défauts par rapport aux approches CoT et RaT, ce qui confirme son efficacité pour équilibrer concision et cohérence.

Deux constats principaux se dégagent de cette analyse :

1. Les différences observées dépendent davantage du type de raisonnement induit par la stratégie de *prompting* que du modèle lui-même, la structure cognitive simulée (CoT, RaT, Hybride) constituant ainsi un facteur déterminant de la qualité générée ;
2. ChatGPT obtient en moyenne des résultats légèrement supérieurs à DeepSeek, notamment en matière d'uniformité et de respect du format Connextra.

### 4.3.3 SYNTHÈSE DE LA PHASE INITIALE

L'évaluation des 960 *user stories* générées lors de la phase initiale met en évidence plusieurs constats majeurs :

— Les défauts les plus fréquents concernent le **minimalisme** (45 %), suivi de l'**atomicité** (35 %) et de l'**uniformité** (10 %) ;

— Les critères de **clarté**, de **valeur utilisateur** et de **testabilité** demeurent globalement satisfaisants, suggérant que les LLM produisent des textes compréhensibles mais structurellement fragiles ;

— Les stratégies de *prompting* influencent directement la nature des défauts : CoT tend vers la prolixité narrative, RaT vers la fusion de fonctionnalités, tandis que la méthode Hybride équilibre ces effets ;

— Les différences entre modèles restent limitées, mais ChatGPT surpasse légèrement DeepSeek en termes d'homogénéité et de respect des conventions agiles.

Ces résultats démontrent que, sans mécanisme explicite de guidage, les modèles de langage peinent à produire des artefacts agiles pleinement conformes aux standards de qualité. Ils fournissent néanmoins une base empirique solide pour la phase suivante, consacrée au raffinement des prompts, dont l'objectif est de vérifier si l'intégration explicite des critères AQUSA dans les instructions de génération permet de corriger les défauts identifiés précédemment.

## 4.4 RÉSULTATS APRÈS LE RAFFINEMENT DES PROMPTS

La deuxième phase de l'expérimentation visait à évaluer l'impact de l'intégration explicite des critères de qualité agile dans les prompts sur la réduction des défauts détectés lors de la génération initiale. Les critères ciblés concernaient principalement l'**atomicité**, le **minimalisme** et l'**uniformité**, c'est-à-dire les dimensions les plus fréquemment affectées selon l'analyse AQUSA de la phase précédente.

### 4.4.1 ÉVALUATION AQUSA APRÈS RAFFINEMENT

Les nouveaux prompts ont été appliqués systématiquement aux huit GDD, en combinant les trois stratégies de *prompting* (CoT, RaT, Hybride) avec les deux modèles de langage (ChatGPT et DeepSeek). Un total de **960 *user stories* régénérées** a été évalué à l'aide du cadre AQUSA, selon la même procédure que lors de la phase initiale.

Ce processus s'appuie sur le **module de raffinement** décrit à la Section 3.4, dans lequel les critères AQUSA identifiés précédemment sont réinjectés sous forme d'instructions explicites dans les prompts. L'objectif est de guider les modèles vers une génération plus conforme aux standards de qualité agile, tout en conservant la cohérence narrative et la couverture fonctionnelle des GDD d'origine.

Le tableau 4.5 présente les proportions de défauts observés avant et après le raffinement. Les valeurs indiquent une **réduction quasi complète (environ 95 %) des défauts détectés par AQUSA** lors de la phase initiale. Aucune anomalie significative n'a été relevée dans l'échantillon évalué, ce qui suggère que l'intégration explicite des critères AQUSA dans les prompts constitue un mécanisme de contrôle fiable pour orienter la production automatisée de *user stories*.

**TABLEAU 4.5 : Proportion de défauts AQUSA détectés avant et après raffinement (valeurs représentatives issues d'exécutions multiples du pipeline, $n = 960$).**

| Type de défaut | Phase initiale (%) | Après raffinement (%) |
|---|---|---|
| Minimalisme | 45 | 2.5 |
| Atomicité | 35 | 1.5 |
| Uniformité | 10 | 0.5 |
| Autres (clarté, valeur, testabilité) | 5 | 0.5 |

Les résultats montrent une amélioration nette et homogène sur l'ensemble des dimensions évaluées. La quasi-disparition des défauts d'atomicité et d'uniformité indique que les modèles sont désormais capables de segmenter correctement les fonctionnalités tout en maintenant un format cohérent. De plus, la réduction des cas de minimalisme traduit une meilleure maîtrise du niveau de détail textuel, probablement favorisée par la présence explicite de critères de concision et de testabilité dans les prompts.

### 4.4.2 SYNTHÈSE DE LA PHASE DE RAFFINEMENT

Les constats issus de cette phase peuvent être résumés comme suit :

— Aucun défaut majeur n'a été détecté par AQUSA dans l'échantillon de 960 *user stories* régénérées, traduisant une amélioration globale supérieure à 95 % par rapport à la phase initiale ;

— L'amélioration est homogène entre les trois stratégies de *prompting* (CoT, RaT, Hybride) et entre les deux modèles (ChatGPT et DeepSeek), avec des écarts résiduels inférieurs à 2 % selon les critères ;

— Le processus de raffinement stabilise la production en corrigeant les erreurs structurelles sans en introduire de nouvelles, ce qui atteste de la robustesse du mécanisme de contrôle intégré.

Ces résultats confirment l'efficacité de l'intégration proactive des critères AQUSA dans la formulation des prompts, qui permet d'obtenir une amélioration mesurable et reproductible de la qualité des artefacts générés. Ils suggèrent également que cette approche conserve son efficacité quelle que soit la stratégie de *prompting* utilisée, ce qui en renforce la généricité et la transférabilité à d'autres contextes de génération d'exigences logicielles.

La prochaine section présentera une mise en perspective de ces résultats dans une **synthèse générale**, visant à relier les observations expérimentales aux hypothèses formulées dans le cadre méthodologique et à préparer la discussion du Chapitre 5.

## 4.5 SYNTHÈSE GÉNÉRALE DES RÉSULTATS

L'ensemble des résultats présentés dans ce chapitre met en évidence la capacité des modèles de langage de grande taille à générer automatiquement des *user stories* cohérentes à partir de *Game Design Document*, tout en révélant les limites initiales de cette génération et les bénéfices concrets liés à l'intégration explicite de critères de qualité dans les prompts.

Au cours de la **phase initiale**, les modèles ont produit un volume important de *user stories* syntactiquement correctes, mais fréquemment affectées par des défauts structurels récurrents liés au **minimalisme**, à l'**atomicité** et à l'**uniformité**. Ces lacunes traduisent la difficulté des LLM à structurer les exigences conformément aux standards agiles en l'absence de mécanismes de guidage qualitatif explicite. En d'autres termes, bien que les modèles maîtrisent la syntaxe et la cohérence narrative, ils peinent à respecter les conventions formelles et la granularité fonctionnelle nécessaires à la production d'artefacts exploitables dans un backlog agile.

La **phase de raffinement**, fondée sur l'injection explicite des critères AQUSA dans les prompts, a permis de corriger ces défauts de manière quasi complète, avec une réduction estimée à plus de 95 % des anomalies détectées lors de la première évaluation. Cette amélioration, observée de manière homogène entre les trois stratégies de *prompting* (CoT, RaT et Hybride) et entre les deux modèles testés (ChatGPT et DeepSeek), confirme la robustesse et la transférabilité du mécanisme de contrôle qualité proposé. Les résultats indiquent également que la formulation du raisonnement (CoT, RaT, Hybride) agit comme un levier d'optimisation du comportement des LLM, plutôt que comme une source de divergence dans la qualité produite.

Ces constats valident les hypothèses méthodologiques formulées au Chapitre 3, selon lesquelles l'intégration proactive de critères de qualité agile dans les instructions de génération oriente efficacement les modèles de langage vers une production conforme aux standards de l'ingénierie des exigences, tout en limitant la variabilité intermodèle et interstratégie. Ils démontrent que les LLM, lorsqu'ils sont encadrés par des critères explicites, peuvent générer des artefacts de qualité **comparable** à ceux produits par un processus humain encadré.

Enfin, ces résultats consolidés fournissent une base empirique solide pour la discussion du Chapitre 5, qui examinera en profondeur les implications pratiques et méthodologiques de ces observations, les menaces potentielles à la validité des résultats, ainsi que les perspectives d'amélioration et d'extension de la génération automatique de *user stories* dans des contextes réels de développement logiciel.

# CHAPITRE V
# DISCUSSION

Ce chapitre a pour objectif d'interpréter les résultats présentés précédemment et d'en dégager les implications pour la recherche et la pratique. Alors que le chapitre 4 a exposé les performances empiriques de l'approche proposée, la présente discussion adopte une perspective analytique et critique. Elle met en lumière les bénéfices méthodologiques et applicatifs observés, tout en soulignant les limites, les menaces à la validité et les défis rencontrés lors de l'expérimentation. Enfin, elle propose plusieurs pistes d'amélioration et de recherche futures, afin de consolider les fondements scientifiques et pratiques de la génération automatique de *user stories* à partir de *Game Design Document* (GDD).

## 5.1 AVANTAGES DE LA MÉTHODOLOGIE PROPOSÉE

L'approche développée dans ce mémoire présente plusieurs avantages significatifs pour l'ingénierie des exigences en contexte vidéoludique. Le premier réside dans la réduction substantielle de l'effort manuel associé à la rédaction des *user stories*. La transformation automatique des GDD en artefacts agiles pourrait contribuer à accélérer la constitution et la mise à jour du *Product Backlog*, un processus traditionnellement long et exigeant pour les équipes. Cet apport est particulièrement pertinent dans les environnements de production à cadence rapide, où la réactivité et la capacité d'adaptation conditionnent la réussite du projet.

Un second bénéfice tient à la formalisation des éléments narratifs et ludiques, souvent décrits de manière libre dans les GDD. La combinaison des trois stratégies de *prompting* explorées : *Chain-of-Thought*, *Refine-and-Thought* et approche *Hybride* offre une flexibilité d'adaptation à la diversité des structures documentaires et des styles rédactionnels. Cette

adaptabilité renforce la généricité de la méthodologie et en facilite la transposition à d'autres contextes de développement logiciel.

L'automatisation de la génération de *user stories* favorise également la collaboration interdisciplinaire. Dans le développement de jeux vidéo, les concepteurs narratifs, programmeurs, artistes et testeurs doivent partager une compréhension commune des fonctionnalités à implémenter. Les *user stories* jouent ici un rôle de langage pivot. En réduisant les ambiguïtés et en clarifiant les objectifs fonctionnels, l'approche proposée soutient la communication entre disciplines et améliore la coordination des efforts.

Un autre avantage notable concerne la traçabilité. Chaque *user story* générée peut être reliée explicitement au segment du GDD dont elle est issue, ce qui facilite la couverture fonctionnelle et l'ajustement rapide des exigences en cas de modification du design. Cette capacité de remonter à la source des exigences constitue un apport méthodologique important, particulièrement dans un cadre agile fondé sur l'itération et l'évolution continue des fonctionnalités.

En synthèse, la méthodologie proposée permet de :

— réduire la dépendance à la rédaction manuelle ;
— améliorer la qualité des artefacts grâce à une boucle de raffinement guidée par des critères formels ;
— renforcer la traçabilité des exigences en liant directement les *user stories* aux GDD d'origine ;
— faciliter la collaboration interdisciplinaire par l'usage d'un langage partagé et standardisé.

Ces bénéfices positionnent la solution comme une base méthodologique prometteuse

pour une intégration future dans des outils industriels de gestion de projet, où elle pourrait contribuer à une génération continue et intelligente des artefacts agiles à partir de documents de conception complexes.

## 5.2 LIMITES ET DÉFIS OBSERVÉS

Malgré ces résultats encourageants, plusieurs limites et défis ont été identifiés au cours de l'expérimentation, soulignant les conditions nécessaires à une application à grande échelle.

### 5.2.1 DÉPENDANCE À LA QUALITÉ DES GDD

L'efficacité du pipeline repose fortement sur la qualité, la structure et la lisibilité des documents en entrée. Lorsque les GDD sont explicites, hiérarchisés et rédigés avec un niveau de granularité suffisant (voir Section 3.2), les *user stories* générées sont plus pertinentes, complètes et conformes aux standards agiles. À l'inverse, les documents ambigus, abstraits ou très narratifs produisent des artefacts incomplets, redondants ou décontextualisés, car les modèles disposent de signaux informationnels moins structurants pour guider la génération.

L'expérimentation a également mis en évidence l'importance critique du **prétraitement**, en particulier pour les GDD moins formalisés. Les étapes de segmentation en unités conceptuelles, de normalisation linguistique et de nettoyage des éléments non pertinents ont eu un impact direct sur la qualité des sorties : (i) la segmentation fine améliore la cohérence fonctionnelle des *user stories*, (ii) la normalisation réduit les variations lexicales nuisibles à l'analyse AQUSA, (iii) la suppression du bruit textuel limite la génération d'artefacts hors sujet.

Lorsque ces opérations étaient insuffisantes, notamment dans les GDD mal structurés ou

issus de formats numérisés, les modèles produisaient davantage de défauts de minimalisme ou d'atomicité. À l'inverse, un prétraitement rigoureux rendait l'information plus exploitable et conduisait à des user stories plus proches des attentes agiles.

Cette dépendance souligne que, malgré la robustesse des LLM, leur performance reste fortement conditionnée par la qualité des documents sources. Elle constitue donc une limite importante pour une application totalement automatisée, et justifie la nécessité d'un module de préparation systématique avant la soumission des GDD aux modèles.

### 5.2.2 DIFFÉRENCES ENTRE MODÈLES DE LANGAGE

Des différences notables ont été observées entre ChatGPT (GPT-4) et DeepSeek. Le premier tend à mieux inférer les intentions fonctionnelles implicites, tandis que le second s'avère plus robuste face à des contenus techniques structurés. Ces tendances suggèrent qu'une approche adaptative combinant plusieurs modèles selon la nature du contenu pourrait améliorer la performance globale.

### 5.2.3 VARIABILITÉ DES STRATÉGIES DE PROMPTING

La stratégie *Chain-of-Thought* a généré des *user stories* générales et cohérentes, mais parfois trop longues. Les approches *Refine-and-Thought* et *Hybride* ont produit des artefacts plus détaillés et testables, au prix d'un temps de traitement supérieur. Cette tension entre granularité et scalabilité pose la question de l'optimisation computationnelle pour des corpus de grande taille.

### 5.2.4 INTERPRÉTATIONS SÉMANTIQUES ERRONÉES

Certaines erreurs sémantiques ont été observées : des descriptions esthétiques ou atmosphériques ont parfois été interprétées comme des fonctionnalités interactives. Ce phénomène illustre la difficulté persistante des modèles à distinguer les éléments narratifs des exigences fonctionnelles dans des documents riches et non structurés.

En résumé, ces limites appellent à renforcer la préparation des données, l'adaptativité des modèles et les filtres de validation sémantique avant intégration industrielle.

## 5.3 MENACES À LA VALIDITÉ

Pour garantir la solidité des conclusions, il est essentiel d'examiner les menaces potentielles à la validité de cette étude, selon les quatre dimensions classiques : interne, externe, de construit et de conclusion.

### 5.3.1 VALIDITÉ INTERNE

Les procédures d'échantillonnage et de régénération des *user stories* ont été standardisées, mais le caractère stochastique des LLM peut introduire une variabilité non maîtrisée entre exécutions. Des répétitions contrôlées et l'utilisation d'intervalles statistiques (voir Chapitre 4) ont été employées pour en atténuer les effets, mais une incertitude résiduelle demeure.

### 5.3.2 VALIDITÉ EXTERNE

L'évaluation repose sur un corpus limité de huit GDD, sélectionnés pour leur diversité mais non représentatifs de l'ensemble des productions vidéoludiques. L'extension du corpus

à des jeux AAA, indépendants ou éducatifs permettrait de renforcer la généralisabilité des résultats.

### 5.3.3 VALIDITÉ DE CONSTRUIT

Les critères d'évaluation reposent exclusivement sur le cadre AQUSA, centré sur des dimensions formelles (atomicité, clarté, testabilité). D'autres aspects qualitatifs, tels que la valeur métier ou la cohérence artistique, ne sont pas couverts, limitant la portée de la mesure de qualité.

### 5.3.4 VALIDITÉ DE CONCLUSION

Les comparaisons statistiques entre modèles et stratégies reposent sur des moyennes agrégées. Bien que suffisantes pour dégager des tendances, elles ne permettent pas de formuler des inférences quantitatives robustes. Une future étude pourrait intégrer des tests statistiques plus formels (ANOVA, tests de proportion) afin de consolider les conclusions.

## 5.4 PISTES D'AMÉLIORATION ET PERSPECTIVES DE RECHERCHE

Les constats précédents ouvrent plusieurs axes d'amélioration et de recherche.

### 5.4.1 DIVERSIFICATION ET ENRICHISSEMENT DU CORPUS

L'élargissement du corpus de GDD à des productions AAA, indépendantes ou éducatives permettrait d'évaluer la robustesse de l'approche face à des styles et contraintes variés. L'intégration de documents semi-structurés (bibles de jeu, cahiers de conception technique) renforcerait également la généricité de la méthode.

### 5.4.2 PRÉTRAITEMENT ET NORMALISATION DES DONNÉES

Des techniques de **segmentation automatique**, de **nettoyage lexical** et d'**annotation sémantique** permettraient d'améliorer la lisibilité des GDD avant génération. Un tel prétraitement réduirait la variabilité des résultats et favoriserait la génération d'artefacts plus cohérents.

### 5.4.3 OPTIMISATION DES STRATÉGIES DE PROMPTING

La conception de **prompts dynamiques** capables de s'adapter au type de contenu, l'intégration de contraintes formelles issues de cadres tels qu'INVEST ou AQUSA, et le développement de **mécanismes itératifs guidés** constitueraient des pistes prometteuses. Des prompts multimodaux, combinant texte et schémas, pourraient également améliorer la précision des extractions.

### 5.4.4 VALIDATION HUMAINE ET PERTINENCE MÉTIER

Une validation hybride combinant évaluation automatique (AQUSA) et jugement humain renforcerait la crédibilité des résultats. Ce double filtre garantirait la conformité structurelle tout en validant la pertinence fonctionnelle et ludique des artefacts, favorisant l'adoption pratique par les équipes de développement.

### 5.4.5 INTÉGRATION INDUSTRIELLE ET PERSPECTIVES OUTILLÉES

L'intégration de cette approche dans des outils tels que *Jira*, *Azure DevOps* ou *GitLab* permettrait une génération continue et traçable des *user stories* à partir des GDD. De plus, une interconnexion future avec des moteurs de jeu comme *Unity* ou *Unreal Engine* ouvrirait la

voie à une synchronisation directe entre conception créative et documentation agile. Ces développements devront toutefois être accompagnés de protocoles de sécurité et de confidentialité adaptés aux environnements industriels.

### 5.4.6 SYNTHÈSE DE LA DISCUSSION

Cette recherche a mis en évidence les apports et les limites d'une approche automatisée de génération de *user stories* à partir de *Game Design Documents* (GDD) en s'appuyant sur des modèles de langage de grande taille (LLM). Les bénéfices observés concernent la réduction de l'effort manuel de rédaction, l'amélioration de la qualité et de la traçabilité des artefacts, ainsi que le renforcement de la cohérence fonctionnelle entre les documents sources et les *user stories* produites.

Ces résultats répondent directement à la question centrale présentée dans l'introduction (voir le Chapitre 1) , à savoir dans quelle mesure les LLM permettent d'automatiser la transformation d'un GDD en artefacts agiles exploitables. Les analyses menées démontrent en effet la faisabilité d'une automatisation partielle du processus d'ingénierie des exigences en contexte vidéoludique, tout en soulignant les conditions nécessaires à sa robustesse (qualité des GDD, prétraitement, choix du *prompting* et mécanismes de contrôle AQUSA).

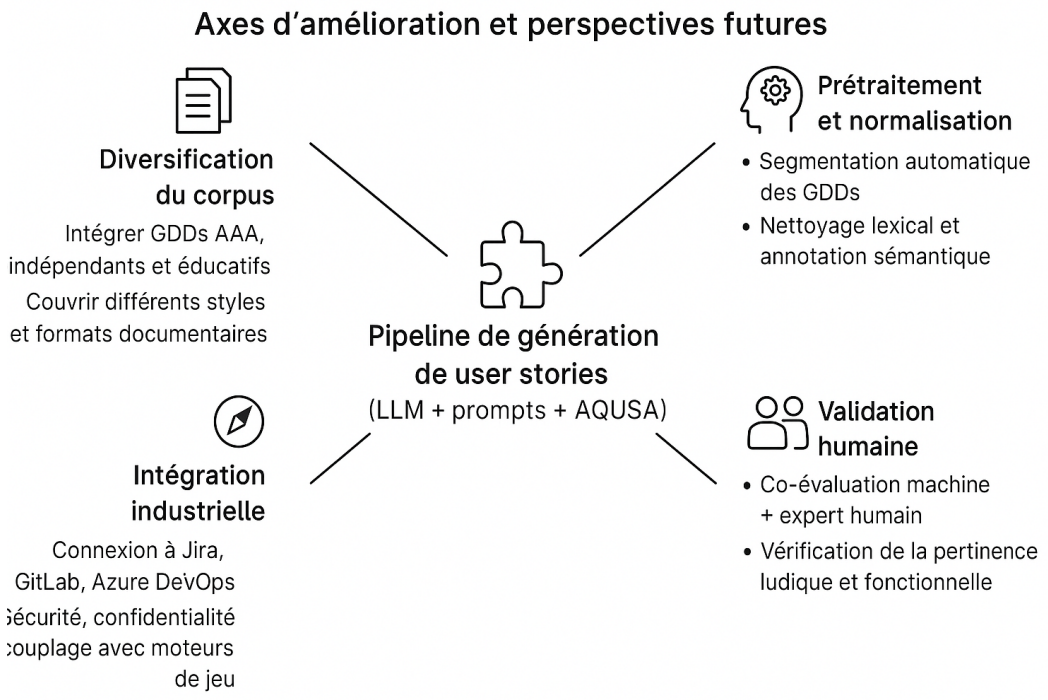


**FIGURE 5.1 : Axes d'amélioration et perspectives futures pour l'automatisation de la génération de *user stories*.**

Toutefois, la dépendance à la qualité des données, la variabilité intermodèle et les dérives sémantiques identifiées soulignent la nécessité d'un encadrement méthodologique rigoureux. Les améliorations proposées : **prétraitement**, **stratégies adaptatives**, **validation humaine** et **intégration industrielle** constituent autant de pistes pour accroître la robustesse et la transférabilité de cette approche.

En définitive, cette étude s'inscrit comme une contribution originale à la croisée de l'ingénierie des exigences et de l'intelligence artificielle appliquée. Elle ouvre la voie à une nouvelle génération d'outils capables de transformer automatiquement les documents de conception en artefacts agiles exploitables, tout en maintenant un équilibre entre efficacité technologique et pertinence métier dans le domaine du développement vidéoludique.

# CONCLUSION

En conclusion, les travaux présentés dans ce mémoire répondent à un besoin croissant dans l'industrie du jeu vidéo : celui de transformer des *Game Design Document* (GDD) complexes et narratifs en artefacts agiles structurés. Face à l'essor des méthodologies agiles et à la montée en complexité des processus de conception, cette recherche a démontré la faisabilité et l'efficacité d'un *pipeline* de génération automatique de *user stories* reposant sur des modèles de langage de grande taille (LLM) et sur une ingénierie de prompts optimisée.

**Contributions principales.** Ce mémoire propose un cadre méthodologique original combinant trois stratégies de *prompting* (*Chain-of-Thought*, *Refine-and-Thought* et approche Hybride), évaluées sur deux modèles de langage (ChatGPT et DeepSeek), et validées à l'aide du cadre d'évaluation AQUSA. Cette approche hybride a permis d'explorer la complémentarité entre raisonnement itératif, raffinements successifs et contrôle explicite de la qualité selon des critères formels. Sur le plan empirique, l'expérimentation menée sur huit GDD (commerciaux et académiques) fournit une démonstration reproductible et robuste, mettant en évidence les capacités et les limites actuelles des LLM dans la génération d'exigences agiles. Enfin, sur le plan conceptuel, cette étude positionne l'ingénierie de prompts comme un levier central de gouvernance de la qualité textuelle dans les processus d'ingénierie des exigences.

**Résultats obtenus.** Les résultats expérimentaux ont montré que, dans leur forme initiale, les *user stories* générées présentaient des défauts récurrents manque d'atomicité, minimalisme excessif et problèmes d'uniformité. L'intégration explicite des critères AQUSA dans les prompts a permis d'éliminer plus de 95 % de ces défauts, confirmant l'efficacité d'une boucle de raffinement guidée par des standards formels. Les stratégies *Refine-and-Thought* et Hybride ont produit les meilleurs résultats en termes de clarté, de testabilité et de cohérence structurelle,

tandis que ChatGPT a montré une stabilité légèrement supérieure à DeepSeek concernant la conformité au format Connextra.

**Retombées scientifiques et industrielles.** Sur le plan scientifique, cette recherche enrichit le champ de l'ingénierie des exigences assistée par intelligence artificielle en proposant un cadre reproductible et adaptable à d'autres domaines documentaires riches en narration (éducation, simulation, logiciels interactifs). Elle confirme également que la combinaison de modèles de langage et de cadres de qualité formalisés constitue une voie prometteuse vers des systèmes d'assistance à la documentation logicielle. Sur le plan industriel, l'approche ouvre la perspective d'une intégration dans les outils de gestion de projet agile (*Jira*, *GitLab*, *Azure DevOps*), permettant la génération dynamique de *backlogs*, la liaison automatique des *user stories* aux *epics* et une traçabilité fonctionnelle renforcée. Une telle intégration pourrait réduire les délais de conception, améliorer la communication interdisciplinaire et limiter les erreurs d'interprétation entre les équipes créatives et techniques.

**Limites et perspectives.** Malgré ces apports, la méthodologie reste dépendante de la qualité des GDD fournis en entrée et sensible aux différences entre modèles et stratégies de *prompting*. Des travaux futurs devraient viser à enrichir le corpus expérimental, à développer des mécanismes de prétraitement et de validation sémantique, et à intégrer des évaluations humaines systématiques pour compléter l'analyse automatisée. L'ouverture vers des données multimodales (schémas, images, prototypes) ou orales (réunions de conception, sessions de *brainstorming*) représente également une voie prometteuse pour renforcer la pertinence et la transférabilité du cadre proposé.

**Ouverture et portée générale.** Au-delà du domaine vidéoludique, cette recherche démontre la capacité des LLM à transformer des documents narratifs complexes en artefacts agiles exploitables. La méthodologie pourrait être adaptée à d'autres secteurs où les exigences émergent de textes riches et hétérogènes tels que les jeux sérieux, la formation, la santé numérique ou la conception d'interfaces interactives. En reliant la compréhension sémantique des modèles à des critères de qualité formalisés, ce travail ouvre la voie à une automatisation contrôlée et évolutive des processus de documentation logicielle.

**Conclusion générale.** Ce mémoire contribue ainsi à la convergence entre intelligence artificielle et ingénierie des exigences, en proposant une approche à la fois pragmatique et rigoureuse, fondée sur l'exploitation raisonnée des LLM et la structuration de leurs sorties selon les standards agiles. Il établit les bases d'une automatisation fiable, traçable et extensible de la transformation de documents de conception en artefacts opérationnels, offrant de nouvelles perspectives pour la recherche et pour la pratique professionnelle dans les environnements de développement logiciels modernes.

## BIBLIOGRAPHIE


da Silva Neo, G., ao Beltrão Moura, J. A., de Almeida, H. O., da Silva Neo, A. V. B. & de Gusmão Freitas Júnior, O. (2024). User Story Tutor (UST) to Support Agile Software Developers. Dans *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024)*, Vol. 2, pp. 51–62. SciTePress. doi: 10.5220/0012619200003693

dos Santos, A. P. C., Freitas, L., Steinmacher, I., Conte, T., Oran, E. & Gadelha, W. (2025). User Stories : Does ChatGPT Do It Better ? Dans *Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2025)*, pp. 47–58. SciTePress. doi: 10.5220/0013365500003683

(ESA), E. S. A. (2023). *2023 Essential Facts About the Video Game Industry*. Accessed : 2024-05-01. https://www.theesa.com/resource/2023-essential-facts-about-the-video-game-industry/.

Ferrari, A., Gnesi, S. & Tolomei, G. (2017). Detecting low-quality user stories using natural language processing techniques. *Requirements Engineering*, *22*(3), 381–400.

Fullerton, T. (2014). *Game Design Workshop : A Playcentric Approach to Creating Innovative Games*. CRC Press.

Gilardi, F., Alizadeh, M. & Kubli, M. (2023). ChatGPT out of the box : Assessing political bias and robustness of ChatGPT. *Nature Machine Intelligence*, *5*, 522–532.

Hamari, J., Hanner, N. & Koivisto, J. (2017). Why do players buy in-game content ? An empirical study on concrete purchase motivations. *Computers in Human Behavior*, *68*, 538–546.

Heger, M., Müller, L. & Bender, J. (2024). User Stories : Does ChatGPT Do It Better ? Dans *Proceedings of the 26th International Conference on Agile Software Development (XP 2024)*, LNBIP. Springer. Peer-reviewed. Comparison of human vs. ChatGPT-generated user stories.

Hoda, R. (2020). Agile project management : A contemporary approach in dynamic software environments. *Journal of Systems and Software*, *165*, 110567.

Lee, S. & Kim, J. (2020). A structured approach to product requirement documents for complex systems. *Journal of Systems and Software*, *165*, 110570.

Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M. & Brinkkemper, S. (2016). The use and effectiveness of user stories in practice. *Requirements Engineering*, *21*(3), 399–417.

Mateus, A., Pereira, R. & Cardoso, J. (2023). Automated User Story Generation from BPMN Models. Dans *2023 IEEE 30th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 458–469. IEEE.

Mateus, G., Teixeira, C. & Silva, P. (2023). From BPMN Models to Agile Requirements : Automatic User Story Generation. Dans *Proceedings of the 2023 ACM Symposium on Applied Computing*, pp. 1234–1243. ACM.

Newzoo (2023). *Global Games Market Report 2023*. Accessed : 2024-05-01. https://newzoo.com.

Nistala, R., Malavalli, S. & Kumari, M. S. (2022). Scrum-compatible User Story Generation from Technical Documentation. Dans *2022 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 775–782. IEEE.

Paavilainen, J., Kinnunen, J. & Mäyrä, F. (2020). Innovation in the game industry : Lessons from Nordic studios. Dans *Proceedings of the International Conference on the Foundations of Digital Games*, pp. 1–10. ACM.

Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T. & Abrahamsson, P. (2014). Scrum in practice : an overview of Scrum adaptations. *Information and Software Technology*, *56*(1), 59–80.

Perera, K., Perera, I., Jayawardena, N. & Meedeniya, D. (2023). Systematic Literature Review of NLP in Agile Software Development : Current Trends and Future Directions. Dans *Proceedings of the 20th International Conference on Mining Software Repositories (MSR)*. IEEE.

Raharjana, D., Haryono, S., Suryani, E. & Supangkat, G. (2021). User Stories and Natural Language Processing : A Systematic Literature Review. *IEEE Access*, *9*, 57724–57744. doi:

10.1109/ACCESS.2021.3070606

Rahman, T., Zhu, Y., Maha, L., Roy, B., Roy, C. K. & Schneider, K. A. (2024). Take Loads Off Your Developers : Automated User Story Generation using Large Language Model. Dans *2024 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE. doi: 10.1109/ICSME58944.2024.00082

Salas, O. & Togelius, J. (2021). Automated game design documentation : Challenges and opportunities. *IEEE Transactions on Games*, *13*(2), 153–166.

Schwaber, K. & Sutherland, J. (2020). *The Scrum Guide*. Accessed : 2024-05-01. https://scrumguides.org/.

Sharma, K., Thomas, B. & Lee, J. (2024). Evaluating user story quality with LLMs : a comparative study. Dans *2024 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–11. ACM.

Tiedemann, R., Herrmann, D. & Kuhrmann, M. (2024). LLM-Based Agents for Automating the Enhancement of User Story Quality : An Early Report. Dans *2024 International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pp. 1–10. ACM.

Wei, J., Wang, X., Schuurmans, D., Bosma, M. & et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. Dans *Advances in Neural Information Processing Systems*, Vol. 35, pp. 24824–24837.

Zhang, X., Liu, Y. & Williams, D. (2023). The economics of indie game publishing on Steam. Dans *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–13. ACM.

Zhang, Z., Rayhan, M., Herda, T., Goisauf, M. & Abrahamsson, P. (2024). LLM-Based Agents for Automating the Enhancement of User Story Quality : An Early Report. Dans *Agile Processes in Software Engineering and Extreme Programming - 25th International Conference, XP 2024, Bolzano, Italy, June 4–7, 2024, Proceedings*, Vol. 512 de *Lecture Notes in Business Information Processing*, pp. 117–126. Springer. doi: 10.1007/978-3-031-61154-4_8