# Text2Stories : Agent IA pour la Génération et l'Évaluation Automatique de *User Stories* à partir d'un Contexte Projet

Marius Ortega[1,2], Hassan Imhah[1], Vanande Khatchatrian[1], Nédra Mellouli[2], Christophe Rodrigues[2], Nicolas Travers[2]

[1]Onepoint, Paris, France

[2]De Vinci Higher Education, De Vinci Research Center, Paris, France

## Contexte et Enjeux

Les **User Stories (US)** sont devenues le standard de l'ingénierie des exigences. Elles suivent généralement la structure : *« En tant que <Acteur>, je veux <Action>, afin de <Valeur> »*. Héritées des méthodes Agiles, elles permettent d'exprimer des exigences fonctionnelles et non-fonctionnelles du point de vue de l'utilisateur final. Leur étude en traitement automatique du langage naturel présente toutefois plusieurs limitations.

**⚠ Données Rares**
Absence de datasets publics combinant US et contexte projet.

**⚠ Audit Informel**
Qualité vérifiée oralement (3 Amigos) sans métriques d'audit.

**⚠ Agilité Distribuée**
Freins méthodologiques au sein des équipes Open-Source.

**Notre Proposition : `Text2Stories` (T2S)**

Une architecture multi-agents modulaire (4 blocs) pour **générer**, **diversifier** et **évaluer** des US incluant Critères d'Acceptation (AC) et Métriques INVEST selon un contexte projet.

## Comparatif avec l'État de l'Art

Les US étant un objet dont la définition exacte peut différer d'un projet à l'autre, les solutions de génération présentes dans l'état de l'art n'intègrent pas nécessairement les mêmes éléments dans leur tâche de génération.

| Fonctionnalité | GeneUS | UStAI | `T2S` (Notre approche) |
|---|---|---|---|
| Génération de **User Stories Contextualisée (US)** | ✗ | ✗ | ✓ |
| Génération de **Critères d'Acceptation (AC)** | ✓ | ✗ | ✓ |
| Évaluation via des **Métriques Standards** | ✗ | ✓ | ✓ |

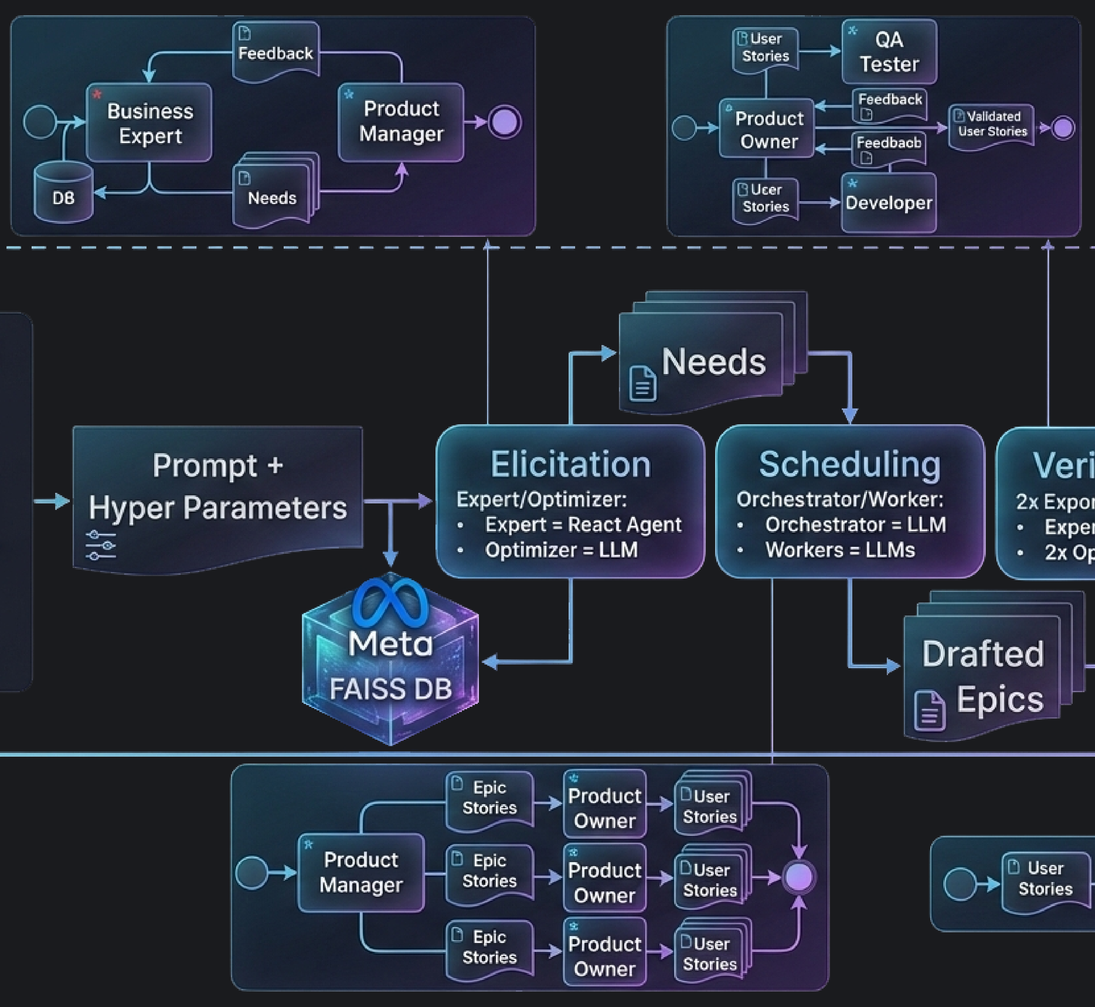
Pour notre usage, nous avons distingué trois critères important dans le cadre de la génération de user stories :

**i) Génération en contexte** : Les US ne sont pas générées seules, mais conjointement avec d'autres et dans le cadre d'un projet spécifié afin de se rapprocher de leur utilisation réelle.

**ii) La génération de critères d'acceptation** pour des exigences testables et vérifiables.

**iii) L'utilisation de métriques standards** (QUS, INVEST) pour l'évaluation et la comparaison des modèles.

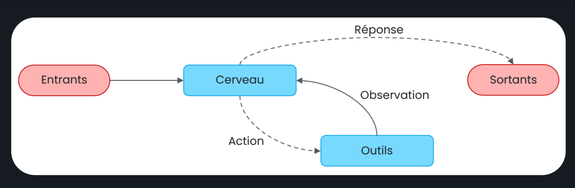
## Le Pipeline Text2Stories




## Architecture Multi-Agents

`T2S` repose sur quatre blocs modulaires simulant une équipe Agile. Ces blocs s'inspirent de sous-agents à l'architecture généraliste introduits par les équipes R&D d'Anthropic. L'usage de sous-agents standardisés permet d'obtenir des modèles plus transparents, maintenables et efficients.
`T2S` combine plusieurs sous-agents avec du Context Management et des Outils customisés (ou Tools) pour LLM.

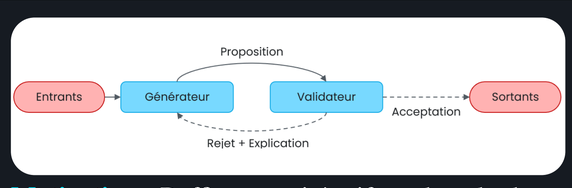
### Expert ReAct




**Motivation :** Exploration itérative et autonome via Raisonnement et appels de Tools.

**Rôle :** Le *Business Expert* interroge la base FAISS itérativement pour la cohérence.

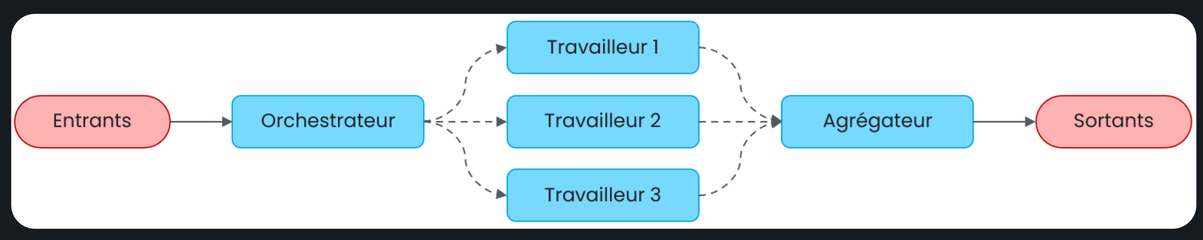
### Expert Optimizer




**Motivation :** Raffinement itératif par boucle de rétroaction guidée par le LLM.

**Rôle :** Le *Product Manager* critique et valide les besoins jusqu'à ce qu'ils soient jugés suffisants.

### Orchestrator-Worker




**Motivation :** Parallélisation asynchrone et orchestrée par le LLM.

**Rôle :** Décompose les *epics* en sous-tâches pour éviter la surcharge cognitive du modèle.

## Résultats et Évaluation

L'évaluation a porté sur 420 US (annotés par 20 consultants Onepoint), comparant `T2S` à des études d'ablation.

- **Alignement Humain (INVEST) :** `T2S` surpasse systématiquement les approches LLM+CoT et RAG+CoT.
- **Unicité :** Le LLM produit 50% de redondances sur des tâches longues, `T2S` garantit une meilleure diversité.
- **Qualité (QUS) :** `T2S` se classe 1er sur la **Minimalité** et la **Forme** des user stories générées.

| Méthode | Alignement Humain-Modèle INVEST | | | | Défauts de Format et d'Unicité AQUSA Score (QUS) | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Accuracy ↑ | Precision ↑ | Recall ↑ | F1-Score ↑ | Atomic ↓ | Well-Formed ↓ | Minimal ↓ | Uniform ↓ | Unique ↓ |
| **LLM** | 0.93 | 0.93 | 1.00 | 0.96 | 0.00 | 0.00 | 0.00 | 0.00 | 0.54 |
| **LLM + CoT** | 0.73 | 0.75 | 0.93 | 0.82 | 0.20 | 0.00 | 0.04 | 0.12 | 0.00 |
| **RAG + CoT** | 0.77 | 0.82 | 0.89 | 0.85 | 0.16 | 0.00 | 0.00 | 0.12 | 0.00 |
| **`T2S` (Gen)** | 0.83 | 0.85 | 0.96 | 0.90 | 0.08 | **0.00** | **0.00** | 0.42 | 0.12 |
| **`T2S` (Deg)** | 0.51 | 0.74 | 0.49 | 0.58 | 0.44 | 0.00 | 0.16 | 0.24 | 0.00 |

## Conclusion & Perspectives

`T2S` offre un cadre robuste pour :

- **Les Product Owners :** Aide au brainstorming et évaluation automatique du backlog.
- **Les Chercheurs :** Génération de jeux de données synthétiques et reproductibles pour le NLP.

**Futur :** Formalisation de la tâche de Génération d'US. Développement de métriques plus finement alignées qu'INVEST et QUS avec la qualité réelle des user stories dans un contexte projet.

**Code Source**

`MariusAAROS/Text2Stories`