

PLAN DE PROJET

Projet Amazon Reviews :

Analyse et classification des avis Clients

BONNAT Jonathan

Data Engineer - B2B-LP-DESFL

INTRODUCTION GENERALE

Ce document contient le plan de projet pour la mise en place du service d'analyse et de classification des avis clients d'Amazon. Il contient une vision claire de l'organisation du projet : planning, livrables, gouvernance, répartition des rôles et conditions de réussite.

La solution technique a été définie et construite dans les projets précédents : extraction des avis depuis PostgreSQL, traitement dans Airflow, classification via un modèle Zéro-Shot, stockage dans S3 et exposition des résultats via une API FastAPI. Ce plan se concentre donc sur la manière dont le projet est structuré et piloté, et non sur la conception technique.

OBJECTIFS DU PROJET

OBJECTIFS FONCTIONNELS

- Identifier automatiquement les avis les plus pertinents laissés par les clients.
- Classer chaque avis dans une thématique (défauts produits, livraison, prix, satisfaction...).
- Mettre ces résultats à disposition d'un mockup e-commerce via une API.
- Permettre aux équipes métier de comprendre rapidement les irritants principaux.

OBJECTIFS TECHNIQUES

- Automatiser l'extraction et le traitement des avis via un pipeline Airflow.
- Intégrer un modèle Zéro-Shot pour la classification.
- Stocker les résultats enrichis dans S3 au format Parquet.
- Exposer les données via une API FastAPI accessible au front.
- Garantir la reproductibilité et la stabilité (Docker, logs, monitoring)

OBJECTIFS ORGANISATIONNELS

- Structurer le projet autour d'un planning clair et réaliste.
- Définir une gouvernance simple : rôles, réunions, validations.
- Anticiper les risques (délais, charges, imprévus techniques).
- Assurer une montée en compétence fluide des futurs utilisateurs.

PÉRIMÈTRE DU PROJET

INCLUS DANS LE PÉRIMÈTRE

Le projet couvre l'ensemble des éléments nécessaires à la mise en place du service d'analyse des avis :

- extraction et transformation des données issues de PostgreSQL ;
- pipeline Airflow pour le traitement et la classification ;
- intégration du modèle Zero-Shot ;
- génération des fichiers Parquet sur le Data Lake S3 ;
- génération de documents Json pour l'historisation
- exposition des résultats via l'API FastAPI ;
- intégration avec le mockup e-commerce fourni ;
- documentation utilisateur et procédures de support.

EXCLUS DU PÉRIMÈTRE

Certaines dimensions ne sont volontairement pas prises en charge dans ce projet :

- développement d'une interface utilisateur avancée (le mockup existant est utilisé tel quel) ;
- création d'un modèle NLP entraîné sur mesure (Zéro-Shot uniquement) ;
- intégration avec les systèmes Amazon internes (non fournis dans le cadre pédagogique) ;
- pilotage d'un cycle complet DevOps CI/CD (hors besoin du projet).

Le périmètre restant permet de se concentrer sur le contenu du service, ainsi que sa documentation et son accompagnement.

PLANNING ET ROADMAP

ORGANISATION GENERALE

Le projet est découpé en cinq phases. La durée totale est de 12 semaines, temps imparti mais suffisant pour un projet mené par une équipe réduite mais complète.

La charge est répartie entre les rôles suivants :

- Chef de projet / Scrum Master
- Data Engineer
- Développeur Fullstack
- Ops
- Product Owner

DÉCOUPAGE PAR PHASES / ROADMAP

| PHASE | PÉRIODE | ACTIVITÉ PRINCIPALE | RÔLES IMPLIQUÉS | CHARGE TOTALE |
|---------------------------|-----------|--|--|---------------|
| Cadrage et préparation | S1 → S2 | périmètre, backlog initial, choix techniques | Chef de Projet Product Owner Data Engineer | 10 JH |
| Développement de l'ETL | S3 → S6 | extraction PostgreSQL, transformations, Parquet S3 | Data Engineer | 20 JH |
| Classification et scoring | S7 → S8 | intégration Zero-Shot, tests, ajustement scoring | Data Engineer Product Owner | 8 JH |
| Déploiement + API | S9 → S10 | FastAPI, Docker, Airflow prod, monitoring | Dev Fullstack Ops | 14 JH |
| Documentation | S11 → S12 | support utilisateur, docs, formation | Tous | 8 JH |

Charge totale : 60 JH

LIVRABLES POUR CHAQUE ÉTAPE

- Phase 1 : périmètre, backlog, document d'architecture cible
- Phase 2 : pipeline ETL en local + premiers outputs Parquet
- Phase 3 : classification Zero-Shot opérationnelle + validation thématique
- Phase 4 : stack Docker, Airflow Prod, API FastAPI, monitoring
- Phase 5 : documentation utilisateur + support

JALONS CLÉS (MILESTONES)

- M1 : Architecture validée → fin S2
- M2 : Pipeline ETL opérationnel → fin S6
- M3 : Classification Zero-Shot intégrée → fin S8
- M4 : API + Airflow en production → fin S10
- M5 : Documentation complète prête → fin S12

GOUVERNANCE DU PROJET

L'objectif de la gouvernance est d'assurer :

- un suivi régulier,
- des décisions rapides
- et une bonne communication;

entre les rôles techniques et le métier (représenté par le PO). Le projet est mené en mode agile, avec des sprints courts (2 semaines) et des livrables fréquents.

ORGANISATION DES RÔLES

Voici les rôles impliqués dans le projet. Certains sont assumés par la même personne, mais les responsabilités restent distinctes.

- Chef de projet / Scrum Master : organisation du planning, animation des rituels, suivi des risques et coordination générale.
- Product Owner (PO) : porte la vision métier, clarifie les thématiques d'avis, valide les priorités du backlog.
- Data Engineer : responsable du pipeline Airflow, de l'intégration Zéro-Shot, des tests et de la qualité des données.
- Développeur Fullstack : responsable de l'API FastAPI et des interactions avec le mockup e-commerce.
- Ops / SRE : configuration Docker, environnement de production, logs, monitoring, disponibilité du système.

Au vu de l'échelle du projet, l'organisation est appropriée. Elle pourra être renforcée si le projet continue et va plus loin.

RITUELS ET INSTANCES AGILES

Le projet est mené en mode agile, mais avec une organisation volontairement simple. L'équipe est réduite et les sujets sont très techniques, donc nous avons retenu uniquement les rituels qui apportent réellement de la valeur au suivi du travail.

- **Daily** (15min chaque matin) :

Chacun indique ce qu'il a fait, ce qu'il va faire, et les éventuels blocages.

- **Backlog Refinement** (30min une fois par semaine) :

Réunion courte pour revoir la priorité des tâches, clarifier certains tickets et préparer le sprint suivant. Le Product Owner y apporte son point de vue et arbitre les besoins métier.

- **Sprint planning** (30min avant chaque sprint) :

Définition des objectifs du sprint et estimation des efforts. La durée du sprint est alignée

avec la roadmap du projet (2 semaines).

- **Sprint Review** (30 min à la fin de chaque sprint) :

Présentation rapide des développements terminés et ajustement selon le reste à faire. Le PO peut en profiter pour faire des retours si besoin.

- **Rétrospective** (30 min après chaque sprint) :

Moment informel qui permet à chaque personne de donner son ressenti personnel sur le sprint écoulé, et définir si besoin des plans d'actions pour ajuster l'organisation

RACI DU PROJET

| ACTIVITÉ | CP | PO | DE | DF | OPS |
|---------------------------|----|----|----|----|-----|
| Définition du périmètre | AR | C | C | I | I |
| Priorisation du Backlog | C | AR | I | I | I |
| Développement ETL | I | I | RA | I | C |
| Zero Shot + scoring | I | C | RA | I | C |
| Développement API + front | I | C | I | R | A |
| Déploiement en prod | C | I | C | C | AR |
| Monitoring et supervision | I | I | C | I | AR |
| Gestion des incidents | AR | I | C | I | R |
| Documentation | AR | C | R | R | I |
| Support Utilisateur | AR | C | C | C | I |

R = Réalise | A = Approuve | C = Consulté | I = Informé

PROCESSUS DE PRISE DE DÉCISION

La prise de décision suit une logique simple :

- le PO valide les priorités métier et la valeur attendue des fonctionnalités ;
- le CP tranche sur les questions de planning, de charges et d'organisation ;
- les rôles techniques (Data Engineer, Fullstack, Ops) décident des choix d'implémentation lorsqu'il s'agit de contraintes techniques ;
- en cas de désaccord, le CP arbitre, puis remonte en comité de pilotage si nécessaire (pas implémenté à l'échelle du projet)

Cela permet de garder un système de décision rapide et proportionné à la taille du projet.

GESTION DES DEMANDES DE CHANGEMENT

Les évolutions (nouvelle catégorie d'avis, ajustement du scoring, modification d'un endpoint API, etc) suivent un cycle léger :

- La demande est formulée par le PO ou détectée par l'équipe technique.
- Le CP évalue l'impact (charge, planning, dépendances).
- Le PO valide l'intérêt métier et la priorité.
- La modification est ajoutée au backlog lors d'un affinage.
- Le changement est intégré dans un sprint si la charge est compatible.

Cette organisation permet d'intégrer les évolutions sans mettre en risque le planning initial.

PRISE EN COMPTE DE L'ACCESSIBILITÉ ET DES SITUATIONS DE HANDICAP

Le projet tient compte des besoins d'accessibilité, conformément aux critères RGAA, pour permettre à n'importe quel utilisateur ou membre de l'équipe de travailler sans problème au quotidien.

Dans la pratique, cela signifie que :

- les documents produits (PDF, guides, slides) respectent des règles simples : contrastes lisibles, police adaptée, structure claire ;
- l'API et les scripts sont documentés de manière textuelle, accessible aux lecteurs d'écran ;
- les ateliers de formation peuvent être menés en mode asynchrone ou avec support vidéo sous-titré ;
- les décisions sont prises en intégrant les contraintes individuelles (rythme, contexte matériel).

SPÉCIFICATIONS FONCTIONNELLES ET TECHNIQUES

Cette section résume les principales spécifications fonctionnelles et techniques du service d'analyse des avis. Elles ont été détaillées dans les documents techniques précédents ; l'objectif ici est de rappeler les points essentiels pour le pilotage du projet.

SPÉCIFICATIONS FONCTIONNELLES

- Le système doit analyser les avis clients présents dans la base PostgreSQL et identifier les avis les plus utiles.
- Chaque avis doit être classé dans une catégorie thématique : défaut produit, livraison, prix, satisfaction, etc.
- Les résultats doivent être consommables par le mockup e-commerce via une API.
- Les équipes métier doivent pouvoir accéder à des données propres, enrichies et cohérentes (score de pertinence, catégorie, métadonnées).
- Les équipes doivent pouvoir également accéder aux données brutes historisées si besoin (par exemple pour les incidents)
- Les traitements doivent être automatisés pour fonctionner quotidiennement sans intervention manuelle.

SPÉCIFICATIONS TECHNIQUES

- Sources de données : base PostgreSQL (avis, produits, acheteurs).
- Pipeline : orchestré par Airflow (extraction, transformation, scoring, génération Parquet).
- Classification NLP : modèle Zero-Shot pré-entraîné, sans phase d'entraînement

locale.

- Stockage : fichiers Parquet sur S3 pour les données enrichies, MongoDB pour l'archivage brut.
- API : FastAPI exposant une route permettant d'obtenir les avis enrichis par produit.
- Conteneurisation : services exécutés via Docker (Airflow, API, environnement d'exécution).
- Monitoring : journaux d'exécution Airflow, monitoring léger sur les conteneurs, logs applicatifs.
- Contraintes : temps d'exécution raisonnable, structure stable des données, résilience des traitements ETL.

FACTEURS CLÉS DE SUCCÈS ET RISQUES PROJET

FACTEURS CLÉS

Les points suivants conditionnent la réussite du projet :

- Qualité du pipeline Airflow : fiabilité des extractions, absence de données manquantes.
- Stabilité du modèle Zero-Shot : cohérence du scoring et pertinence des catégories d'avis.
- Simplicité de l'API : endpoints clairs, rapides, et compatibles avec le mockup e-commerce.
- Collaboration PO ↔ Data Engineer : indispensable pour ajuster les catégories et valider la valeur métier.
- Documentation accessible : guides clairs, formats adaptés (contrastes lisibles, structure simple).
- Bon pilotage agile : rituels courts mais réguliers, décisions rapides, backlog bien tenu.

RISQUES PROJET

Chaque risque est suivi : lors du comité opérationnel hebdomadaire (pas nécessaire à l'échelle du projet), le CP l'évalue et met à jour un registre simple (niveau, action, responsable).

- Variabilité des avis (longueur, langue, bruit) pouvant réduire la qualité du Zéro-Shot.
- Temps d'exécution Airflow si le volume d'avis augmente trop rapidement.
- Dépendance à plusieurs rôles techniques : disponibilité limitée, charge forte du Data Engineer.
- Risque d'incompréhension métier si les catégories NLP ne sont pas validées régulièrement par le PO.
- Problèmes d'intégration API si la structure des données change en cours de route.
- Accessibilité insuffisante si la documentation n'est pas adaptée à tous les utilisateurs (RGAA).

CONCLUSION

Ce document détaille comment le service d'analyse et de classification des avis clients est organisé, piloté et livré. Il propose une structure simple, adaptée à une équipe réduite, tout en respectant les principes d'un pilotage agile et les exigences du référentiel.

Le cadrage, la roadmap, l'organisation de l'équipe et les rituels agiles permettent de garantir un avancement régulier et lisible. Les choix techniques (Airflow, Zéro-Shot, S3, API FastAPI) sont maîtrisés et ont déjà fait leurs preuves dans les blocs précédents. Les principaux risques sont identifiés et suivis, et la prise en compte de l'accessibilité permet d'assurer une utilisation efficace pour tous les utilisateurs.

Ce document sert de référence pour la suite du projet et facilitera la transition vers les phases de maintenance, de support utilisateur et d'évolution. Il pose une base solide pour assurer la continuité du service et son intégration dans un environnement opérationnel.