Document de Gestion des Risques – Plateforme SaaS de Surveillance et d'Aide à la Décision

5.1 Introduction

Ce document identifie, analyse et propose des stratégies de mitigation pour les risques liés au projet :
Plateforme SaaS de surveillance et d'aide à la décision basée sur les avis Google Play + Chatbot IA + Intégration Jira/Confluence.

Les risques sont classés par catégorie :

- **Techniques**
- **Organisationnels**
- **Opérationnels**
- **Sécurité & données**
- **IA / Modèles**
- **Scraping & réglementation**
- **SaaS / Production**

Chaque risque est évalué sur :

- **Probabilité (P) : Faible / Moyenne / Élevée**
- **Impact (I) : Faible / Moyen / Élevé**
- **Criticité = P × I**

5.2 Tableau synthétique des risques

ID	Catégorie	Risque	P	I	Criticité	Stratégie
R01	Technique	Scraping bloqué par Google	Élevée	Élevé	⚠️ Critique	Rotation IP, proxy, délais, fallback API
R02	Technique	Changements du HTML Play Store cassent le scraper	Moyenne	Élevé	⚠️ Critique	Tests automatiques & monitoring
R03	IA	Précision insuffisante des modèles NLP	Moyenne	Élevé	⚠️ Critique	Fine-tuning + dataset propre
R04	IA	Modèle trop lourd → latence élevée	Moyenne	Moyen	Moyen	Optimisation ONNX / quantization
R05	Données	Données bruitées / spam dans les avis	Élevée	Moyen	Moyen	Filtrage NLP + heuristiques

R06	Sécurité	API non sécurisée → fuite de données	Faible	Élevé	⚠️ Critique	Auth JWT + rate limit + HTTPS
R07	Organisationnel	Mauvaise répartition des tâches → retards	Moyen	Moyen	Moyen	Planification agile + daily meetings
R08	Infra	Surcharge CPU/mémoire sur scraping	Moyenne	Moyen	Moyen	Scalabilité horizontale
R09	SaaS	Mauvais dimensionnement Multi-tenant	Faible	Élevé	Moyen	Analyse avant développement
R11	Frontend	Dashboard trop lent (trop de requêtes API)	Moyen	Moyen	Moyen	Caching / pagination
R12	Backend	Mauvaise conception API (pas scalable)	Faible	Élevé	⚠️ Critique	Architecture microservices
R13	Production	Mauvaise configuration SSL / CORS	Moyen	Élevé	⚠️ Critique	CI/CD + tests automatiques
R14	RGPD	Données personnelles non conformes	Faible	Élevé	⚠️ Critique	Audit RGPD + anonymisation
R15	SaaS	Downtime lors du scraping intensif	Moyen	Moyen	Moyen	File queues (Celery/Redis)
R16	UX	Dashboard trop complexe → adoption faible	Moyen	Moyen	Moyen	UX testing early
R17	Chatbot	Mauvaise compréhension des requêtes utilisateurs	Moyen	Moyen	Moyen	Dataset d'intents + feedback loop
R18	Monitoring	Absence de logs détaillés	Faible	Moyen	Faible	Intégration Sentry + Grafana
R19	Atlassian API	Limitation / permissions API Jira / Confluence	Moyenne	Moyen	Moyen	Test sandbox, gestion permissions, logs

---

5.3 Analyse détaillée des risques critiques

🔴 **R01 — Le scraping Google Play peut être bloqué**

Problème : Google peut bloquer l'IP, limiter les requêtes ou modifier sa page HTML.
Probabilité : Élevée | Impact : Élevé | Criticité : Critique
Solutions :

- **Mise en place de délais anti-spam (sleep)**
- **Rotation d'IP / proxies**
- **Scraping distribué (Cloud Functions)**
- **Cache interne pour pages déjà scrappées**
- **Fallback vers API tierce (non officielle mais stable)**

🔴 **R02 — Changement structure HTML → scraper cassé**

Problème : Un changement mineur du DOM du Play Store peut casser le pipeline.
Solutions :

- **Tests automatiques du scraper**
- **Monitoring + alertes en cas de réponse anormale**
- **Architecture modulaire du scraper**
- **Correction rapide grâce à logs détaillés**

🔴 **R03 — Modèles IA peu précis**

Problème : Sentiment analysis & classification de bugs peuvent échouer avec sarcasme, avis courts ou hors sujet.
Solutions :

- **Fine-tuning sur dataset spécifique**
- **Data augmentation**
- **Validation croisée**
- **Indicateurs qualité (F1-score > 80%)**

🔴 **R06 — Faille sécurité API → fuite de données**

Problème : API exposée publiquement → risques d'injections, accès non autorisé, spam.
Solutions :

- **Auth JWT avec expiration**
- **Rate limiting**
- **HTTPS obligatoire**
- **Audit sécurité OWASP**

🔴 **R13 — Mauvaise configuration SSL / CORS lors du déploiement**

Problème : Bloque le frontend → "Blocked by CORS policy".
Solutions :

- **Reverse proxy NGINX bien configuré**
- **CI/CD vérifiant les headers**
- **Tests Postman automatisés**

5.4 Plan de mitigation général

A. Prévention

- **Architecture propre et modulaire**
- **Documentation détaillée**
- **Tests unitaires et intégration**
- **Méthodologie Agile (sprint de 2 semaines)**
- **Daily meetings**
- **Revues de code systématiques**

B. Détection rapide

- **Logs centralisés**
- **Monitoring (Sentry, Grafana)**
- **Alertes e-mail pour scraper/IA/backend**
- **Tests automatisés à chaque commit**

C. Correction / Contingence

- **Rollback instantané via Docker**
- **Versioning API**
- **Mode "dégradé" si IA tombe → statistiques basiques**
- **Mode manuel si scraping tombe → import CSV**

5.5 Matrice des risques

Impact vs Probabilité :

IMPACT

FAIBLE MOYEN ÉLEVÉ

ÉLEVÉ | R01 R02 R06 R13

MOYEN | R11 R17 R03 R12 R19

FAIBLE | R15 R18 —

FAIBLE MOYEN ÉLEVÉ

PROBABILITÉ

5.6 Conclusion

La majorité des risques critiques sont techniques (scraping, IA, sécurité).
Avec les stratégies de mitigation mises en place, l'équipe peut conserver un niveau de risque global maîtrisé. L'intégration Jira/Confluence est plus stable et moins critique que l'Odoo initialement prévu.