Découpage de projet – Plateforme de Surveillance et d'Aide à la Décision (Version Jira/Confluence)

Chaque phase comprend :
- **Objectifs**
- **Acteurs impliqués**
- **Livrables**
- **Outils**
- **Risques**

Phase 1 — Cadrage & Architecture

Objectifs
- **Définir le périmètre fonctionnel du projet.**
- **Concevoir l'architecture globale : frontend, backend, pipeline de scraping, module IA, intégration Jira/Confluence.**
- **Identifier les besoins métiers et les cas d'usage.**
- **Planifier la répartition des tâches et le calendrier.**

Acteurs impliqués
- **Chef de projet / Architecte**
- **Lead Frontend**
- **Lead Backend**
- **Data/NLP Engineer**
- **Scraping Engineer**

Livrables
- **Cahier des charges complet**
- **Diagrammes : architecture, workflow, flux de données**
- **Maquettes UI/UX initiales**
- **Planning global et répartition des tâches**
- **Document de gestion des risques**

Outils
- **Notion / Confluence (documentation)**
- **Figma (maquettes)**
- **Miro / Draw.io (diagrammes)**
- **Jira / Trello / GitHub Projects (gestion de projet)**

Risques
- **Mauvaise définition du périmètre → retards en cascade**
- **Sous-estimation du scraping (blocages Google)**
- **Choix technologique inadapté difficile à corriger**

Phase 2 — Fondations Techniques

Objectifs
- **Initialiser les environnements frontend, backend, IA et scraping.**
- **Créer la structure de code et configurer la base de données.**

• Mettre en place l'authentification et l'API de base.
• Configurer Docker pour harmoniser les environnements.

Acteurs impliqués
• Backend Developer
• Frontend Developer
• DevOps / Fullstack
• Architecte technique

Livrables
• Projets initiaux (/frontend, /backend)
• Base de données opérationnelle
• Docker + docker-compose
• Authentification JWT ou OAuth
• API CRUD minimale
• Documentation Swagger / OpenAPI

Outils
• Backend : Node.js/Express ou Python FastAPI/Django
• Base de données : PostgreSQL ou MongoDB
• Infra : Docker, Docker Compose
• Frontend : Next.js + TailwindCSS
• CI/CD : GitHub Actions

Risques
• Problèmes de configuration CORS
• Incompatibilités d'environnement et dérives de versions

Phase 3 — Scraping & Pipeline de Données

Objectifs
• Collecter automatiquement les avis Google Play.
• Gérer volumes, pagination et filtres temporels.
• Nettoyer, structurer et stocker les données pour l'IA.

Acteurs impliqués
• Scraping Engineer
• Backend Developer
• Data Engineer

Livrables
• Module Scraper (Python)
• Script d'extraction automatisée
• Pipeline ETL (collecte → nettoyage → insertion en base)
• Cron job ou tâche planifiée
• Documentation technique

Outils
• Python, Scrapy / google-play-scraper, Playwright

• Pandas, SQLAlchemy
• Celery + Redis pour tâches planifiées

Risques
• Blocage du scraping par Google
• HTML dynamique → scraper cassé
• Données bruyantes ou incohérentes

Phase 4 — IA : Analyse de Sentiments & Classification

Objectifs
• Développer le module d'analyse IA : sentiment (positif/neutre/négatif) et thèmes techniques.
• Fournir un score de priorité pour l'aide à la décision.
• Intégrer les modèles via API backend.

Acteurs impliqués
• Data Scientist
• NLP Engineer
• Backend Developer

Livrables
• Modèles IA entraînés
• Service IA exposé via API
• Base enrichie avec sentiment, thème et score
• Notebooks d'analyse et validation
• Documentation technique

Outils
• HuggingFace Transformers, PyTorch, scikit-learn, spaCy
• ONNX Runtime pour optimisation
• Jupyter Notebook

Risques
• Modèles trop lourds → lenteur
• Précision insuffisante → besoin de données supplémentaires
• Overfitting → modèle non généralisable

Phase 5 — Tableau de Bord & Interface Utilisateur

Objectifs
• Développer le dashboard et les pages de visualisation.
• Afficher KPIs : sentiment, évolution, top problèmes, note moyenne.
• Garantir une UX optimale et responsive.
• Intégrer le backend via API.

Acteurs impliqués
• Frontend Developer
• UX/UI Designer
• Backend Developer (support API)

Livrables
- **Dashboard complet : sélection application, analyse globale, vue détaillée, chatbot**
- **Filtres (dates, sentiments, thèmes)**
- **Documentation frontend**

Outils
- **Next.js, TailwindCSS, React Query, Axios**
- **Recharts / Chart.js / ECharts**
- **Figma pour design final**

Risques
- **UX mal optimisée**
- **Trop de requêtes API → lenteurs**

Phase 6 — Chatbot & Intégration IA

Objectifs
- **Créer un chatbot conversationnel pour interroger la plateforme.**
- **Permettre des questions métier (ex : "Quels sont les problèmes fréquents ?").**
- **Gérer le contexte et l'historique des conversations.**

Acteurs impliqués
- **NLP Engineer**
- **Backend Developer**
- **Frontend Developer**

Livrables
- **Chatbot intégré au dashboard**
- **API conversationnelle**
- **Modèle de compréhension des intentions**
- **Gestion du contexte (vector DB)**

Outils
- **LangChain / Rasa / Transformers**
- **ChromaDB ou Pinecone**
- **FastAPI (serveur chatbot)**
- **Socket.io pour temps réel**

Risques
- **Mauvaise compréhension des requêtes complexes**
- **Latence si modèle non optimisé**

Phase 7 — Intégration Jira / Confluence

Objectifs
- **Connecter la plateforme aux API Jira pour créer tickets automatiques.**
- **Utiliser Confluence API pour documenter automatiquement les rapports et analyses.**
- **Synchroniser les données via API/backend.**

Acteurs impliqués
- **Backend Developer**
- **Chef de projet**

Livrables
- **Module Jira / Confluence intégré**
- **Formulaires et vues personnalisées dans Confluence (si nécessaire)**
- **Automatisation tickets Jira selon priorité**
- **Documentation utilisateur et technique**

Outils
- **API REST Atlassian (Jira / Confluence)**
- **Python / Node.js backend**
- **Requests / Axios**

Risques
- **Limites API Atlassian**
- **Permissions ou authentification mal configurées**

Phase 8 — Tests, Sécurité & Validation

Objectifs
- **Garantir stabilité, performance et sécurité.**
- **Tester tous les modules : scraping, IA, API, UI, intégrations Atlassian.**
- **Valider le pipeline complet.**

Acteurs impliqués
- **QA Engineer**
- **Backend Developer**
- **Frontend Developer**
- **Data Engineer**

Livrables
- **Suite de tests automatisés**
- **Tests API (Postman)**
- **Tests unitaires et d'intégration**
- **Rapport sécurité (auth, sécurité API)**

Outils
- **Pytest, Jest, Postman / Newman**
- **OWASP ZAP, Sentry**

Risques
- **Tests IA difficiles (résultats probabilistes)**
- **Failles API (JWT mal configuré)**

Phase 9 — Déploiement & Livraison

Objectifs
- **Déployer la plateforme complète : backend, frontend, IA, scraping et intégrations Jira /**

Confluence.
- **Configurer monitoring et scalabilité.**
- **Assurer une livraison stable et documentée.**

Acteurs impliqués
- **DevOps Engineer**
- **Backend Developer**
- **Frontend Developer**

Livrables
- **Version en production**
- **CI/CD opérationnel**
- **Monitoring complet**
- **Documentation finale**
- **Formation utilisateur**

Outils
- **Docker / Docker Compose**
- **Railway.app / AWS / Azure / Render**
- **NGINX, GitHub Actions**
- **Grafana + Prometheus, Sentry**

Risques
- **Mauvaise configuration SSL**
- **Surcharge CPU lors du scraping ou IA**
- **Déploiement non reproductible → rollback difficile**