**Gestion de Projet**

**1 - Identification et Interactions des Parties Prenantes**

**Parties Prenantes Identifiées**

* **Client Final** : Responsable de l’utilisation du produit fini et principal bénéficiaire.
* **Équipe de Développement** : Développeurs backend, frontend, et ingénieurs en données.
* **Chef de Projet** : Responsable de la coordination, de la planification et de la gestion des risques.
* **Équipe de Tests (QA)** : Vérifie la qualité, la performance et l’ergonomie des fonctionnalités.
* **Équipe de Sécurité** : Garantit la conformité avec les normes RGPD et effectue des audits de sécurité.
* **Utilisateurs Pilotes** : Fournissent des retours lors des tests en conditions réelles.
* **Partenaires Externes** : Fournisseurs d’API tierces et prestataires d’hébergement cloud.

**Interactions**

* Réunions hebdomadaires avec toutes les parties prenantes pour assurer le suivi.
* Points de contrôle mensuels avec les clients pour valider les livrables.
* Communication continue via Slack ou Microsoft Teams.
* Documentation partagée sur Confluence ou Google Workspace.

**2 - Portée du Projet**

**2.1 - Énoncé**

Le projet vise à concevoir et développer une application d’analyse et de gestion des données clients. Il inclut des fonctionnalités telles que la segmentation client, la prévision des tendances, et la détection des anomalies. L’application doit être évolutive, sécurisée, et compatible avec divers environnements, tout en garantissant une conformité stricte aux normes RGPD.

**Objectifs clés :**

1. Automatiser les analyses et rapports pour faciliter la prise de décision.
2. Fournir des outils d’IA avancés pour prédire les comportements clients.
3. Permettre l’intégration avec des systèmes tiers (CRM, ERP).
4. Assurer un déploiement cloud évolutif avec haute disponibilité.

**2.2 - Tableau Pas Ceci/Cela (Cours Partie 3)**

| **Pas Ceci** | **Mais Cela** |
| --- | --- |
| Un outil d’analyse statique. | Une plateforme dynamique et interactive avec tableaux de bord personnalisés. |
| Un système rigide. | Une solution modulaire et évolutive grâce aux microservices. |
| Un traitement manuel des rapports. | Des rapports automatisés et exportables dans divers formats. |
| Une gestion centralisée des données. | Une approche décentralisée via des services API REST. |

**3 - Planification**

**3.1 - Étapes**

**3.1.1 - Construire une Stratégie de Risque**

1. **Identification des Risques :**
   * Pannes techniques ou interruptions de service.
   * Vulnérabilités de sécurité non détectées.
   * Délais dans la livraison des fonctionnalités critiques.
2. **Évaluation des Risques :**
   * Analyse de probabilité et d’impact.
   * Attribution d’un niveau de gravité (faible, moyen, élevé).
3. **Réponses aux Risques :**
   * Plans d’atténuation : surveillance des logs, tests de charge.
   * Plans de contingence : procédures de restauration rapide en cas de défaillance.
4. **Suivi et Contrôle :**
   * Révisions hebdomadaires avec documentation des risques identifiés et résolus.

**3.1.2 - Calendrier de Projet**

| **Étape** | **Description** | **Durée Estimée** |
| --- | --- | --- |
| Phase 1 : Conception | Création des maquettes et architecture initiale. | 1 mois |
| Phase 2 : Développement | Implémentation des modules et tests internes. | 3 mois |
| Phase 3 : Tests | Tests fonctionnels, d’intégration et de performance. | 1 mois |
| Phase 4 : Déploiement | Mise en production et suivi initial. | 1 mois |
| Phase 5 : Maintenance | Support continu et améliorations. | Permanent |
|  |  |  |

**3.2 - Outils**

**3.2.1 - Gestion de Projet et Collaboration**

* **Jira :** Suivi des tâches, sprints et gestion des backlogs.
* **Confluence :** Documentation partagée et mise à jour en temps réel.
* **Slack/Microsoft Teams :** Communication instantanée et réunions virtuelles.
* **GitHub/GitLab :** Gestion du code source et intégration CI/CD.

**3.2.2 - Tests et Suivi des Performances**

* **JMeter :** Tests de charge et de stress.
* **Postman :** Tests des API et validation des réponses.
* **Grafana et Prometheus :** Monitoring des performances en temps réel.

**3.2.3 - Sécurité**

* **OWASP ZAP :** Détection des vulnérabilités et tests d’intrusion.
* **ELK Stack :** Suivi des logs pour diagnostiquer les incidents.