



Feuille de Route

Architecte Logiciel
Wiam el yadri



TABLE DES MATIERES

1	CONTEXTE	2
2	OBJECTIFS DE CE DOCUMENT	2
3	PARTIES PRENANTES ET ROLES	3
3-1	MATRICE DES PARTIES PRENANTES	4
4	ARCHITECTURE CIBLE	5
5	ROADMAP REP'AERO	5
5-1	CHOIX ET PRIORITES	5
5-2	ROADMAP	8
5-3	LES COMPETENCES	9
5-4	LA REPARTITIONS DES TACHES ET CONSTITUTION DES EQUIPES	10
5-5	L'AGILITE AVEC LA METHODE SCRUM	10
6	INTEGRATION CONTINUE	11
7	LIVRAISON CONTINUE	13



1 CONTEXTE

Rep'Aero est une entreprise du sud-ouest de la France. Elle travaille comme sous-traitant de maintenance des pièces d'avion pour des compagnies aéronautiques qui opèrent sur des flottes d'avions de transport commerciaux ou business.

Elle travaille avec différents fournisseurs chez qui elle commande des pièces détachées, qu'elle suit et gère et suivre avec un outil de gestion de fournisseurs.

Les pièces commandées sont gérées à l'aide d'un outil de gestion des stocks.

Elle s'occupe d'effectuer des missions de maintenance auprès des clients à l'aide d'un outil de gestion des clients qui permet de garder le contact.

Et pour avoir une trace de différentes interventions, elle utilise un outil de *gestion et suivi* de maintenance du Domaine Production.

L'infrastructure IT existante jugée lente et peu fonctionnelle est l'une des causes de la perte d'un client important.

Dès lors, des objectifs stratégiques à atteindre ont été définis :

- Maintenir les services actuels
- Proposer de nouveaux services
- Recentrer les capacités de l'entreprise

Une architecture cible a été élaboré par un prestataire, ainsi qu'un audit technique rapide de l'infrastructure existante.

Par conséquent la société Rep'Aero souhaite voir évoluer son infrastructure existante vers l'architecture cible, donc un projet de migration est envisagé.

2 OBJECTIFS DE CE DOCUMENT

La feuille de route du projet est un aperçu, de haut niveau granulaire des composants vitaux de tout projet. Il s'agit d'une ressource non confidentielle et accessible au public que les chefs de projet peuvent partager avec un groupe public ou professionnel pour expliquer succinctement ce qui suit :

- Un résumé des objectifs du projet.
- Principaux livrables du projet.
- Dépendances des tâches du projet.
- Jalons cruciaux du projet.
- Risques du projet.



- Le planning du projet
- Les dépendances et liens entre les différentes tâches et livrables.

3 PARTIES PRENANTES ET RÔLES

Tableau N 1: Les Parties Prenantes et leurs rôles

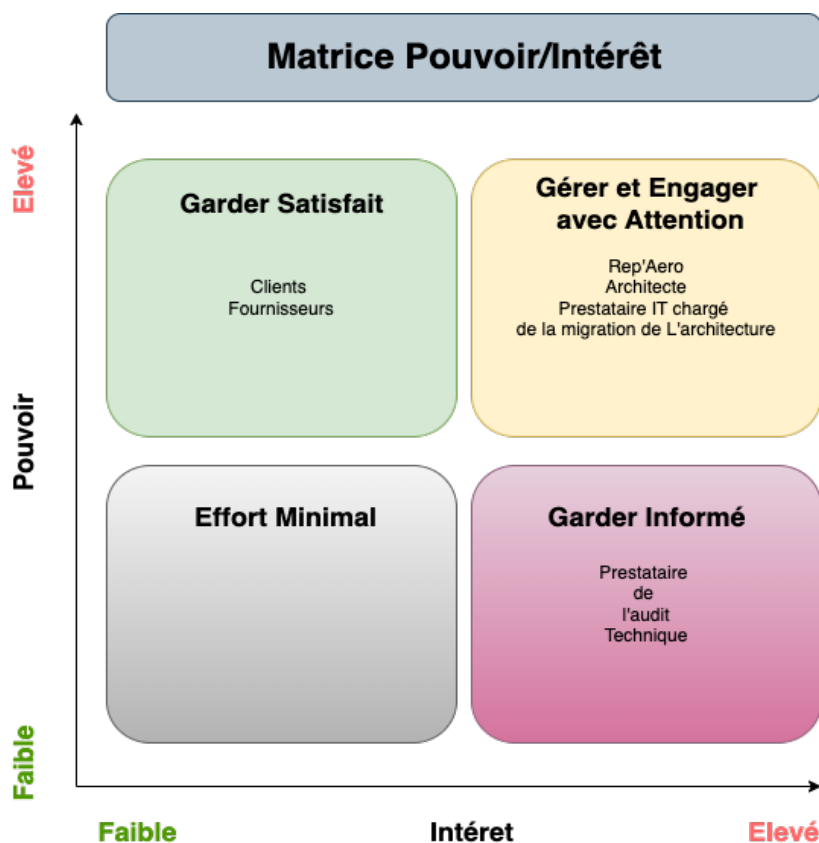
Parties Prenantes	Rôles
Cadres et techniciens de Rep'Aero	<ul style="list-style-type: none">• Responsable des recettes fonctionnelles en collaboration avec le chef de projet.• Chargé d'accompagner les équipes de développement dans la compréhension métier.• Autorités bien définies pour la réalisation du plan
Les Fournisseurs Rep'Aero	Satisfaction sur la qualité du service
Prestataire de service informatique	<ul style="list-style-type: none">• Développements• Déploiement de la nouvelle architecture.
Les Clients Rep'Aero	<ul style="list-style-type: none">•Satisfaction sur la qualité du service•Service reçu en temps voulu•Écoute attentive sur les nouveaux besoins



Prestataire de l'audit technique	<ul style="list-style-type: none"> • Réalisation d'un Audit technique sur l'architecture existante
Prestataire de l'architecture cible	<ul style="list-style-type: none"> • Réalisation de l'architecture cible • Information sur l'avancée du projet
L'architecte	Responsable de : <ul style="list-style-type: none"> • L'étude de faisabilité • La Roadmap • Leu plan d'implémentation. • La préconisation des technologies nécessaires pour une migration optimale

3-1 MATRICE DES PARTIES PRENANTES

Utilisation de la matrice : dresser la liste les parties prenantes, puis les positionner dans la matrice ci-dessous selon leur pouvoir d'influencer la stratégie et leur intérêt à le faire. La matrice pouvoir/intérêt aide à la formulation d'une stratégie compatible avec les attentes des parties prenantes.





4 ARCHITECTURE CIBLE

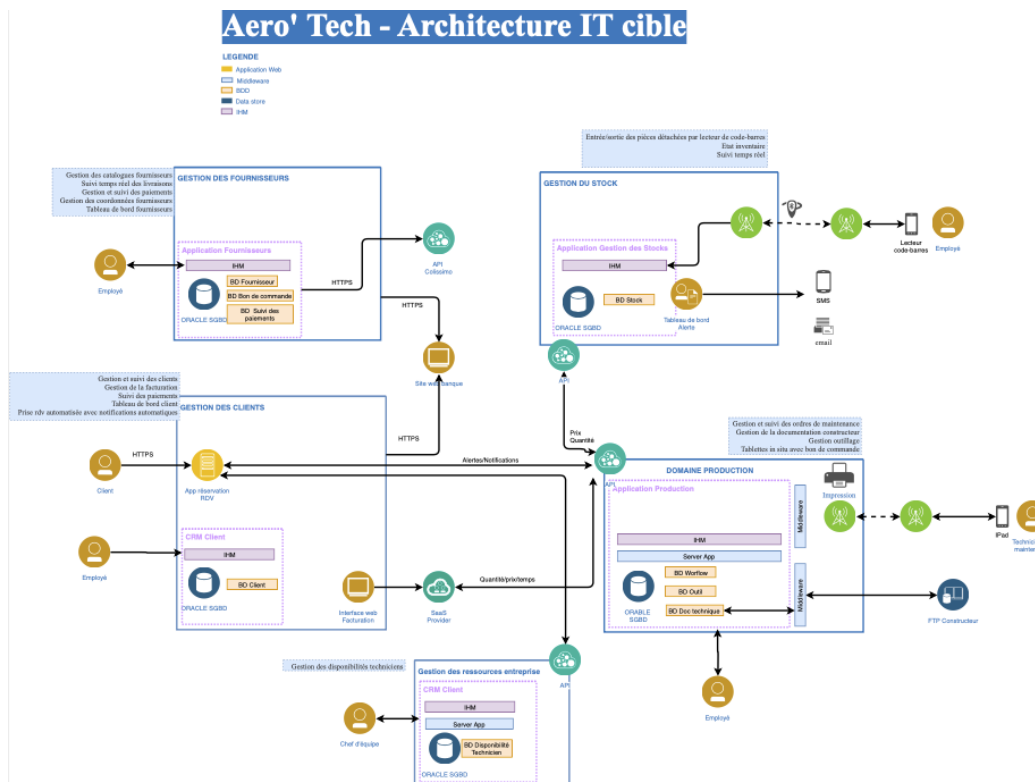


Figure N1 l'architecture Cible

5 ROADMAP REP'AERO

5-1 CHOIX ET PRIORITES

La Roadmap réalisée répond aux choix et priorités suivants :



Les Jalons	Les parties prenantes	Les services	Les Enjeux
1	<ul style="list-style-type: none"> • Gagner la confiance des collaborateurs • Rassurer, Remotiver les collaborateurs • Proposer d'autres services aux clients 	<ul style="list-style-type: none"> • Proposer d'autre service aux collaborateurs • Amélioration des processus de prévision et de gestion des stocks • Le suivi en temps réel des livraisons • Maintenir les services existants • Proposer d'autres nouveaux services • Entrées/sortie des pièces détachées par lecture du code barre 	<ul style="list-style-type: none"> * Gain de temps de traitement en supprimant les opérations manuelles sous EXCEL * Automatisation dans le suivi des stocks. * Création et envoi des alertes liées à la gestion des stocks.



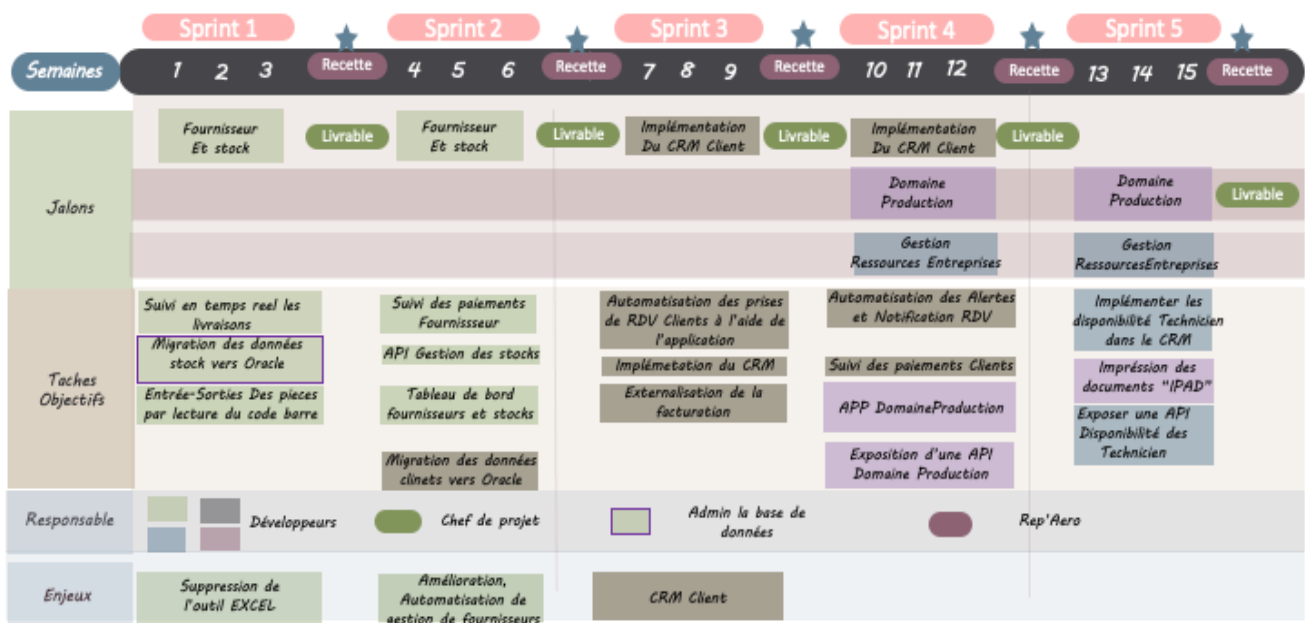
2	<ul style="list-style-type: none"> • Rendre l'expérience de client plus attractive • Faciliter l'interaction entre le système et le client • Automatiser le processus de prise de rendez vous 	<ul style="list-style-type: none"> * Automatiser le processus de prise de rendez vous * Le choix de créneau * La disponibilité de Technicien * Maintenir les services existants * Proposer de nouveaux services * Fournir une meilleure interopérabilité avec des systèmes externes. * la création d'une application web de réservation R.D.V * Faciliter la gestion et le suivi des clients par les employés de l'entreprise Rep'Aero. * Un système basé sur le C.R.M 	<ul style="list-style-type: none"> * Réduire le temps de traitement des demandes de clients * Réduire le temps pour les interventions Techniques
3	<ul style="list-style-type: none"> * Offrir une vision claire sur l'organisation des équipes * Optimiser le temps de travail grâce à l'accompagnement des équipes 	<ul style="list-style-type: none"> * Une meilleure exploitation du C.R.M * Mettre en place un module de gestion des ressources 	<ul style="list-style-type: none"> * Accompagner les équipes afin d'optimiser le temps de travail



		entreprise	
4	<ul style="list-style-type: none"> * Offrir une maintenance de qualité * une gestion de l'ordre de planning pour la maintenance optimisée. * Aider les employés pour faciliter leurs tâches 	<ul style="list-style-type: none"> * Offrir une maintenance de qualité * Une automatisation des mises à jour * Amélioration de la gestion de l'outillage * Mettre en place des nouveaux services * Accès à la documentation 	<ul style="list-style-type: none"> * Offrir une maintenance de qualité

5-2 ROADMAP

IT roadmap Rep'Aero





5-3 MIGRATION DU SYSTEME ACTUEL

Les Tâches Requises pour la migration de l'architecture existante vers la cible

Tableau N 2 : Migration des données

TÂCHES Requises	Profils	Technologie	Description
La migration module Fournisseurs	Développeur	A déterminer	Choix Technologique à déterminer (Langage, Framework)
La migration module gestion des stocks	Développeur	A déterminer	Choix Technologique à déterminer (Langage, Framework)
La migration module Gestion des clients	Développeur	A déterminer	Choix Technologique à déterminer (Langage, Framework)
La migration Domaine de Production	Développeur Admin Data	CRM à déterminer	Choix Technologique à déterminer (Langage, Framework)
La migration Module RH	Développeur		Un chef de projet sera nécessaire tout au long du projet. Un U.X design pour maquetter les écrans des applications futures. Des Ingénieurs système seront nécessaires Pour assurer le déploiement de



			l'architecture cible et des environnements de travail.
--	--	--	---

5-4 LA REPARTITIONS DES TACHES ET CONSTITUTION DES EQUIPES

Les équipes de développement ont la responsabilité de tester la migration pour s'assurer des résultats obtenus, avant de l'importer dans l'environnement de QA, et par la suite l'importer à l'environnement de production.

5-5 L'AGILITE AVEC LA METHODE SCRUM

Dans la méthode agile de gestion de projet, Scrum utilise des sprints comme intervalles de temps pendant lesquels l'équipe va réaliser un certain nombre de tâches du backlog. L'intervalle de temps dépend des besoins de l'équipe, et pour notre cas on a besoin de deux à trois semaines est une durée qui nous permet d'avoir une visibilité sur les fonctionnalités implémentées.

- Les équipes se rencontrent tous les jours pour le "**Daily Scrum**", une réunion quotidienne où chacun fait part aux autres de son avancement. On appelle également ces réunions des "**Daily Stand-Ups**".
- Chaque sprint se termine avec une **Rétrospective**, qui réunit toute l'équipe afin de partager les retours d'expérience et discuter des améliorations possibles du prochain sprint.
- Mettre en place une pratique Devops, pour automatiser un certain nombre de processus.

La solution incrémentielle au fur et à mesure des sprints.

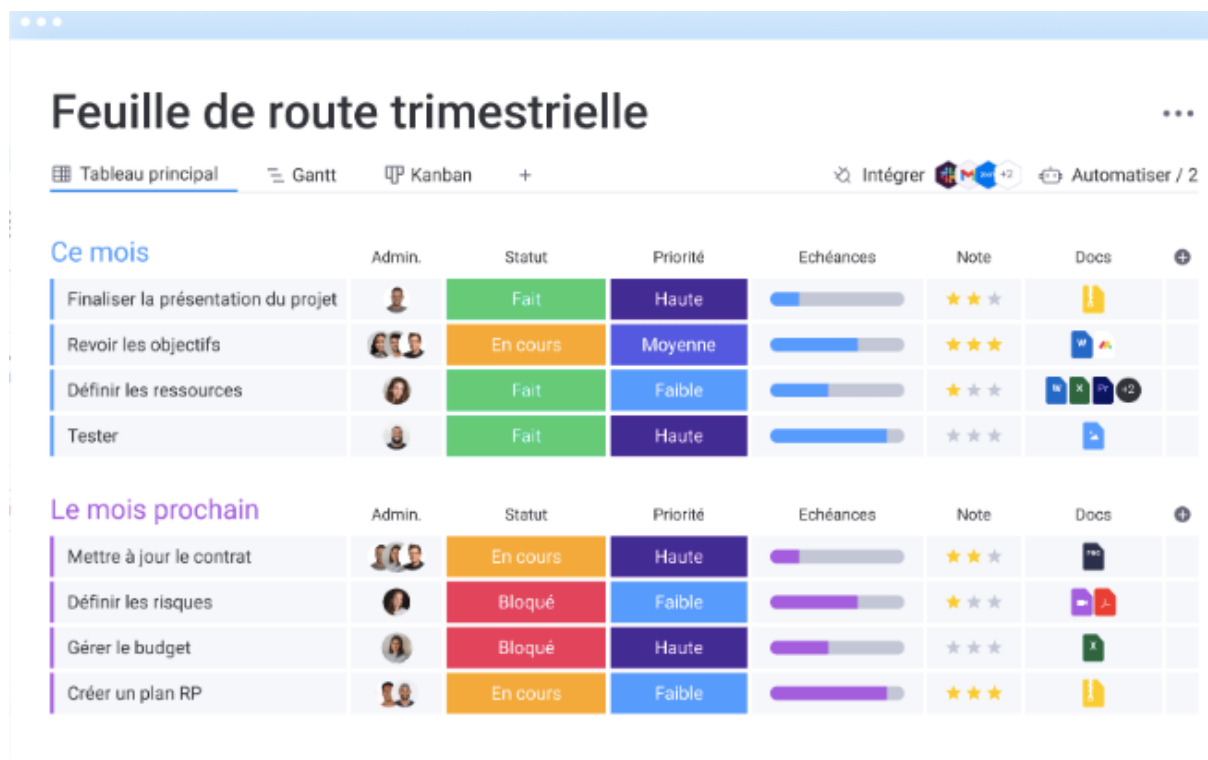


Figure N 2 la feuille de route Rep'Aero

L'utilisation d'outil de gestion de projet comme Monday permet de Créer un espace de travail collaboratif complet pour l'équipe Rep'Aero, sur une plateforme flexible, pour que chacun puisse suivre le projet facilement.



6 INTEGRATION CONTINUE

L'intégration continue est un ensemble de pratiques qui consiste à ce que les développeurs intègrent régulièrement leurs modifications de code à un référentiel centralisé afin de s'assurer que ces modifications ne produisent pas de régression dans l'application développée.

Des outils permettent de réunir les étapes nécessaire à la mise en place de l'intégration

Les développeurs intègrent régulièrement leurs modifications de code à un référentiel centralisé.

L'intégration continue va se faire en 5 étapes :

- **Planification de développement.**



- **Compiler** et intégrer le code.
- **Tester le** code.
- Mesurer la **qualité** du code.
- Gérer les **livrables** de l'application.

Les outils qui seront utilisés pour notre projet seront :



GitHub est une plate-forme d'hébergement de code pour le contrôle de version et la collaboration. Il vous permet, à vous et à d'autres, de travailler ensemble sur des projets où que vous soyez.



JUnit est un framework de test unitaire pour le langage de programmation Java

Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires et de non-régression.

Docker pour le packaging.



Il est possible de compiler, déployer, voire le créer un Artefact (livrable) avec l'intégration continue. Cependant nous pouvons aller bien plus loin, nous pouvons carrément créer un conteneur Docker directement depuis le CI.

L'objectif de faire construire une image Docker directement depuis le CI est d'automatiser à 100% le processus de packaging de l'application. Manuel représente == plus de risque.



En automatisant au maximum les opérations de packaging, nous pourrions par la même occasion de réduire les risques :

- D'erreurs.
- De corruption du livrable.
- De perte de temps.

SonarQube pour afficher des rapports sur la qualité et l'évolution du code.



- Il vérifie que notre base de code est propre et maintenable, pour augmenter la vitesse des développeurs
- Il corrige les vulnérabilités qui compromettent l'application
- Il détecte les bogues délicats pour empêcher un comportement indéfini d'avoir un impact sur les utilisateurs finaux.

7 LIVRAISON CONTINUE

La livraison continue est une méthode de développement de logiciels dans le cadre de laquelle les modifications de code sont automatiquement préparées en vue de leur publication dans un environnement de production à n'importe quel moment.

Elle reste la suite logique de l'intégration continue. Dans l'intégration continue, nous cherchons à ce que le code compile bien, mais aussi qu'il soit fonctionnel en production et de qualité, en lançant le plus régulièrement possible les tests unitaires. Mais il existe d'autres types de tests (au-delà des tests unitaires).

Il peut s'agir de tests d'interface, de charge, d'intégration, de fiabilité de l'API, etc., tout aussi importants, pour garantir la qualité du code. Ces tests ne peuvent cependant pas être lancés sans avoir un environnement déployé.

En suivant cette méthode de livraison continue les développeurs peuvent vérifier les mises à jour et détecter les bugs à corriger avant le déploiement.

Mise en place un pipeline CI/CD

Il est préconisé d'avoir de mettre en place 4 environnements bien cloisonnés à savoir :

- Un environnement réservé aux développeurs :



Un environnement qui permet d'effectuer des tests et de détecter éventuellement des régressions ou anomalies. Il est à disposition des testeurs et des équipes métiers. Il permet d'effectuer des démonstrations lors des livraisons de fin de sprint et par la suite des réajustements.

- Un environnement de pré-production (User Acceptance Test) :

Un environnement qui permet de se rapprocher le plus de la production.

Il permettra d'effectuer les tests de charge, des tests de performance, tests utilisateurs.

- Un environnement de production :

Il permet de planifier et communiquer les livraisons dans ces environnements, et attribués des accès réservés aux personnes concernés par cette étape de projet.

TDD "Test-Driven Développement"

Une technique de développement logiciel qui vise à réduire les anomalies d'une application en favorisant la mise en œuvre fréquente de tests. Une politique de test first, en cohérence avec les méthodes agiles, qui pousse les programmeurs à faire preuve de plus de rigueur et à corriger en quasi-temps réel les bugs et autres erreurs dans le code source.

L'implémentation de la solution passe par l'écriture des différentes fonctionnalités qui débouche sur des scénarios puis des tests (TDD).

L'utilisation de framework comme Cucumber permet faire 'une introduction à l'automatisation de scénarios avec Cucumber en Java

Planning Prévisionnel

L'implémentation de la solution est découpée en 4 sprints de 2 à 3 semaines.

Chaque sprint est ponctué par une démonstration, une livraison en environnement de Test et des ateliers de formation pour les utilisateurs.

(Voir le Roadmap)