# KWANZEO

# KWANZEO

**Data.experts** : créé par des spécialistes, forte croissance

**Data.share** : nous partageons notre passion au travers d'évènements, de publications, d'avis d'experts.

**PDG Hubert de Charnacé** : ingénieur et entrepreneur (MCNEXT - Infeeny, ESN de 350 p.), passionné d'informatique et d'électronique

**DG Marie-Laure Anthonioz** : historienne et directrice marketing et communication. Spécialiste du partage du savoir

# KWANZEO

**Data.intelligence** : améliorez, repensez, transformez vos outils décisionnels. Bénéficiez des dernières solutions et du cloud.

**Data.management** : adoptez un Data Hub / MDM pour une vision unique des données de référence.

# KWANZEO ——— Projets.Azure.data

## Sprint.0

- Définition de la bonne architecture
- Implémentation de ce que les développeurs doivent avoir pour commencer à développer

## Autres sprints

Planification - Conception - Développement - Évolution de l'infrastructure si besoin - Écriture et réalisation des tests - Déploiement - Revue

# KWANZEO —— Goto.Azure.data.sprint.0

- Définir **l'architecture** infrastructure et applicative & la sécurité
- Créer les plateformes de **développement / recette** voir production
- Instancier Azure Data Factory / Stockage
- Éditer **les bonnes pratiques** du projet
- Créer les **templates** Data Factory / les nomenclatures objets BDD
  & les répertoires du Data Lake
- Définir la stratégie des **tests** techniques / non régression et fonctionnels
- **Use case** : récupération des uses cases du 1er sprint

# KWANZEO —— Goto.Azure.data

## Sprint.0.risques | Lenteur<br>Non Qualité<br>Surcoût | **Impact majeur tout au long du projet**

- Infrastructure "faite à la main" :
  - **risque de sécurité , de disponibilité et de non réutilisabilité**
- Absence ou faiblesse des définitions **des bonnes pratiques** du projet
- Absence ou faiblesse **des templates Data Factory** fournis aux équipes
- Absence ou faiblesse dans la définition du cadre de **Test**
- **Lenteur, perte de temps et d'argent** dans la réalisation du Sprint 0
- **Hétérogénéité et problèmes** dans le travail de l'équipe durant tout le projet
- **Manque de fiabilité des mises en production à terme**

**Constat : les besoins et les risques sont les mêmes partout.**

KITKAD

YOUR AZURE DATA FIRST SPRINT

# Sprint.0.succès et plus

- **Déploiement (et redéploiement) automatique de l'infrastructure Data**
- **Déploiement continu automatisé**
- **Usine logicielle pour la création des pipelines ADF (flux de données)**
- **Tests automatisés pour plus de qualité**

---

Accélération du projet / Adaptabilité / Fiabilisation / Bonnes pratiques

KITKAD

YOUR AZURE DATA FIRST SPRINT

# Infrastructure Data

Le design de **l'architecture d'infrastructure** est **sensible**.

Il faut prendre en compte entre autre les aspects :

- **Fiabilisation** des déploiements et des mises en production futures
- **Authentification** applicative (Droits entre éléments Azure )
- Mise en place **des logs et audits** de sécurité nécessaires à la surveillance applicative
- Chaque plateforme (Dev, recette, production) ***doit être équivalente***.

KITKAD

YOUR AZURE DATA FIRST SPRINT

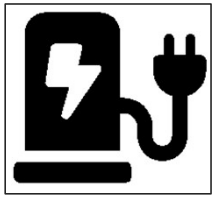
# 1.Déploiement infra Data

Création automatique

des plateformes **développement - recette -production**

[Azure Data Factory - Database SQL DB - Comptes de stockages - Secrets Key Vault (chaînes de connexion) - Azure Functions - Comptes managés - Mécaniques de mise en œuvre de l'audit et logs -…]

**Azure DevOps : ordonnanceur d'étape. Terraform pour l'infra As Code**
**Sizing DB, Nomenclature adaptée au contexte spécifique**



**Votre Infra Azure Data prête à l'emploi, redéployée à chaque Build**

KITKAD
YOUR AZURE DATA FIRST SPRINT

# 2.Déploiement continue

**Compilation** du code / 1er jeux de **tests techniques**
(exécution des flux de données ) : y a-t-il une erreur technique majeure ? Arrêt de la livraison

## Plateforme de **recette**

**Déploiement de l'infrastructure** (As a code) permettant d'avoir des plateformes équivalentes : le code fourni par les Dev fonctionnera sur les autres plateformes
**Déploiement des codes** : Azure functions / base de données / Azure Data Factory
**Test Automatique Technique et Fonctionnel** : correspondance aux attentes fonctionnelles

## Plateforme de **production**

**Déploiement de l'infrastructure** (As a code)
**Déploiement des codes**

**Logique CI / CD automatisation via des scripts Azure DevOps prêts à l'emploi**

KITKAD

YOUR AZURE DATA FIRST SPRINT

# Azure Data Factory

ADF est essentiel car il gère **les flux de données** provenant de **multiples sources**. ADF gère les flux au travers de « **pipelines** ».

**Il existe des risques de dégradation, de lenteur en développement lors de l'évolution du code :**

- si le code Azure Data Factory a été **non factorisé** (une modification impacte plusieurs endroits dans le code au lieu d'un seul)
- lors de l' **ajout d'une source de donnée, s'il y a un manque d'abstraction** (**pas de template**, **pas variabilisation du paramétrage**, « codage en dur »)

**Les bonnes pratiques recommandent l'usage de templates ADF (usine logicielle).** Le risque de performance dégradée en développant des templates ADF non éprouvés chez d'autres clients existe !

KITKAD

YOUR AZURE DATA FIRST SPRINT

# 3.Usine logicielle Azure Data Factory

- Factorisation du code via la **création de pipelines paramétrées**
- Création d'une **pipeline** principale appelant **les autres pipelines** avec les bons paramètres
- **Table de paramétrage** indiquant la pipeline à utiliser et décrivant les paramètres
- **Fourniture des pipelines** permettant la copie à partir des sources de données de type : compte de stockage Azure / ftp / ftps (csv, parquet, json, xml ) , api . Utilisation simplifiée via la table de paramétrage.

KITKAD

YOUR AZURE DATA FIRST SPRINT

# 3.Usine logicielle Azure Data Factory

**KitKAD fournit une usine logicielle pour ADF : Objectif « low code »**

**Accélération et rationalisation** de la partie Data Factory par le **déploiement de scripts** prenant en charge **les flux de copie et l'ordonnancement des transformations**. Utilisation **de tables de paramétrage**. La logique de transformation est à écrire (Store proc, data flow, databricks…).

---

## KitKAD : Usine logicielle pour ADF

## Création rapide de pipelines performants !

KITKAD

YOUR AZURE DATA FIRST SPRINT

# Tests automatisés

**Tester c’est essentiel !**

- Les tests de non régression sont parfois non faits ou faits partiellement car trop longs à effectuer à la main.
- La liste des tests de non régression augmente au fil des sprints du projet.

KITKAD

YOUR AZURE DATA FIRST SPRINT

# Tests automatisés

La mise en place de **tests automatisés** dès le démarrage du projet permet :

- d'éviter un effet « **mur de description** » des **tests de régression** si ces tests sont décrits à posteriori
- de garder **un temps constant** dédié au test
- **de fiabiliser les mises en production** car les tests sont faits et validés

...

KITKAD

YOUR AZURE DATA FIRST SPRINT

# 2.Tests automatisé

## KitKAD automatise les tests !
## Amélioration de la qualité, gain de temps



- Exécution de tous les flux (trigger ADF) sur un jeux de données
- Récupération de la valeur dans la base de données après passage des flux
- Vérification des valeurs dans les tables (comparaison à un attendu)

KITKAD

YOUR AZURE DATA FIRST SPRINT

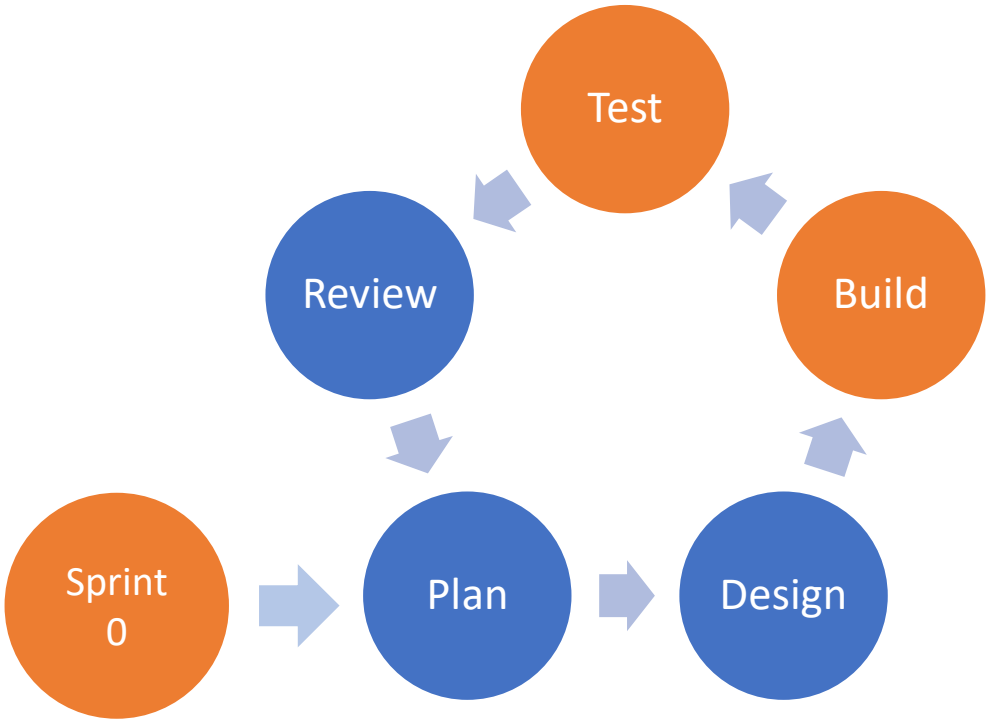
# Le Starter-Kit

**Déployez** facilement la plateforme Azure Data. Gagnez du temps sur **ADF**. Pensez **CI/CD et test automatisé** dès le début

- **Déploiement** ou redéploiement des plateformes (**développement, test et production**) via une logique d'infra As code
- **Intégration et déploiement continu par l'exécution** de scripts Azure DevOps prêts à l'emploi
- **Usine logicielle ADF : Accélération et rationalisation** de la partie Data Factory par le **déploiement de scripts** prenant en charge les flux de copie et d'ordonnancement des transformations. Utilisation de tables de paramétrage.
- Automatisation des **tests pour plus de qualité et de rapidité**

KITKAD
YOUR AZURE DATA FIRST SPRINT

# Agilité




Sprint 0 :
- Mise en place de l'architecture
- Mise en place des pipelines CI/CD
- Mise en place du moteur des tests auto
- Mise en place de l'usine logicielle ADF

=> **Réduction du temps de 80%**

Build :
- Usine logicielle ADF permettant une mise en place des activités de copie et un ordonnancement des transformations rapides

=> **Réduction du temps de 20%**

Test :
- Gestion des tests auto inclus dans les pipelines CI/CD (description des tests dans un fichier Excel)

=> **Réduction du temps de 90%**

KITKAD

YOUR AZURE DATA FIRST SPRINT

# Le Starter-Kit

## KitKAD est adaptable. Le code est fourni.

## Kwanzeo vous accompagne

- Prise en comptes de la nomenclature client
- Prise en compte des règles de gestion du client pour les tests automatisés
- Livraison des codes afin de permettre une adaptation au besoin client
- Transfert de propriété dans le cadre d'un usage interne (groupe et ses filiales)

# Offre

- **Coût** : 10000 €HT + intervention d'un Consultant en régie (minimum 5 jours)
- **Contenu du Package :**
  - Scripts : IaC / Pipeline CI/CD / Codes (ADF, Azure Functions, Test Auto)
  - Documents d'exploitation (mise en place nouvelle plateforme / utilisation de l'usine logicielle ADF / Description de tests automatisés)

**Contenu d'intervention :**

- 2 ateliers infras / métiers
- Adaptation du KitKAD pour le client (Scope : nomenclature / sizing des éléments PaaS / compte manager)
- Mise en place du KitKAD (3 plateformes (dev /recette /prod) et Pipeline CI/CD)
- 1er Flux de copie / 1er test auto
- 1 journée de formation pour l'équipe de dev / DevOps / rédacteur test auto

# KWANZEO

**contact@kwanzeo.com**

**12, rue des Jeûneurs 75002 PARIS | 01 88 33 86 27**