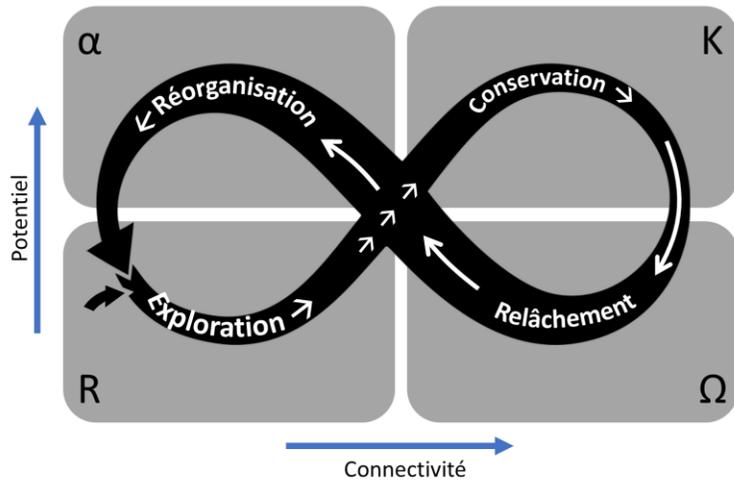
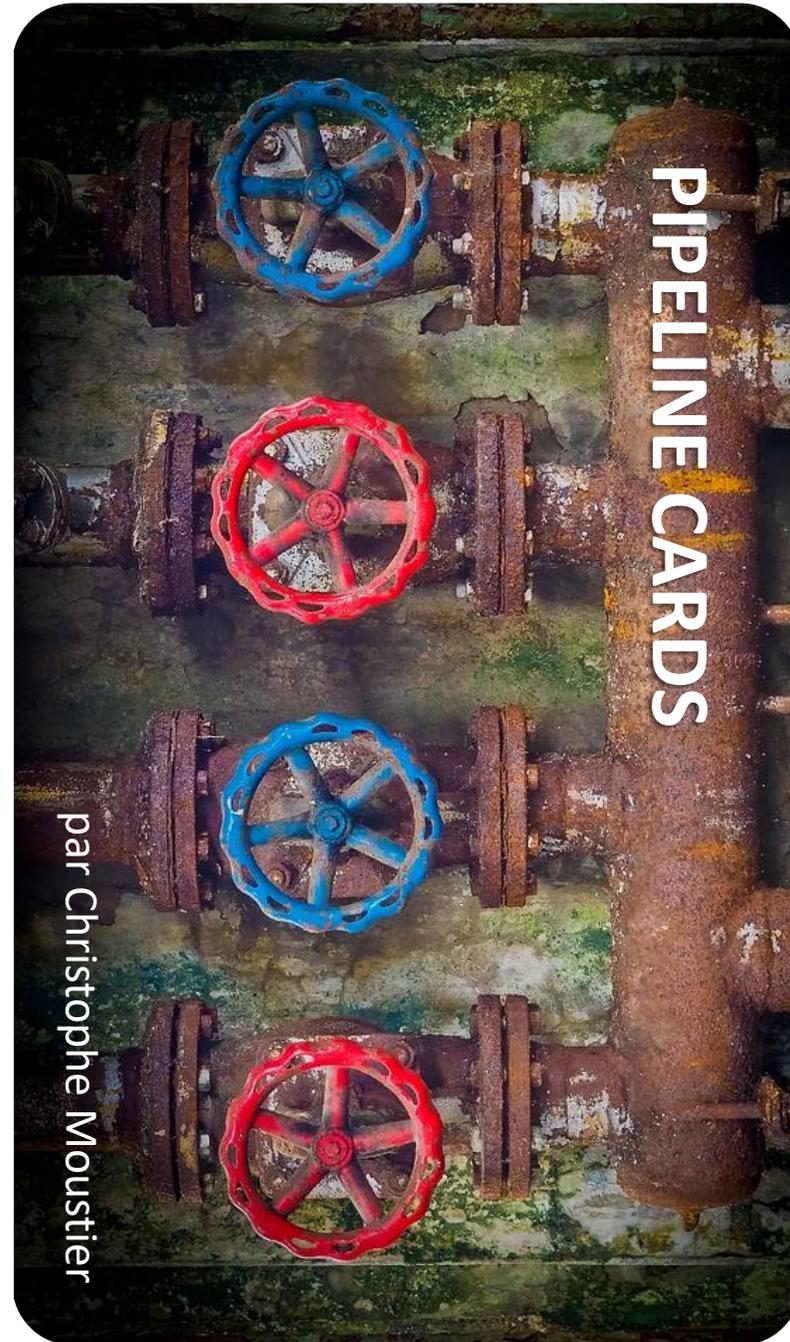


# PIPELINE CARDS



par Christophe Moustier



## Licensing

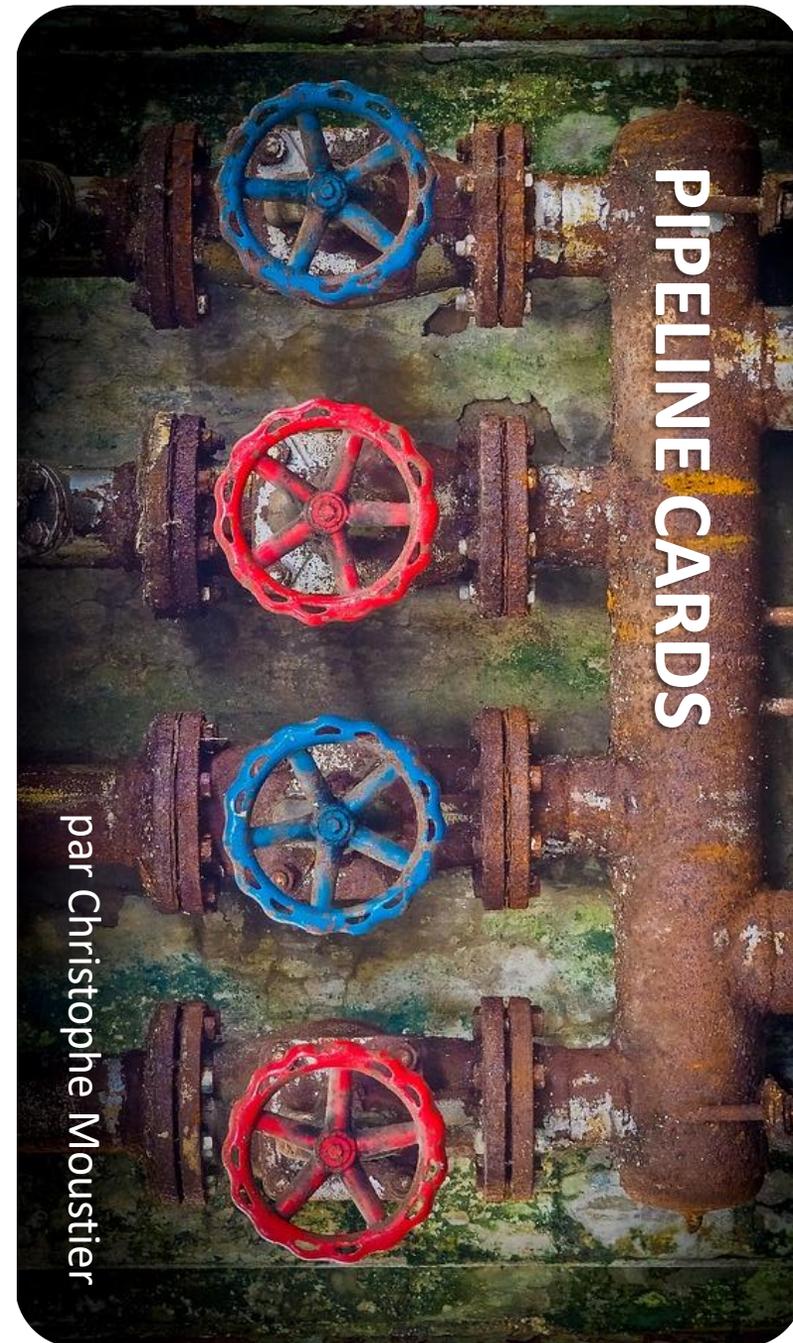
Ce jeu de carte est sous licence CC-BY 4.0. Vous pouvez librement l'utiliser, à la condition de l'attribuer à l'auteur en citant son nom

<https://creativecommons.org/licenses/by/4.0/legalcode.fr>



### Credits

- Icons: monkik (<https://www.flaticon.com>)
- Image PipelineMichaelGaida 2017  
<https://pixabay.com/fr/photos/endroits-perdus-usine-vieux-2178884/>



## PIPELINE CARDS

Jeu de cartes pour développer votre pipeline DevOps et la pratique du test agile mis à l'échelle par le modèle du « Pantesting ».

Il est composé de 5 familles

- *Build*
- *Environnement*
- *Solution*
- *Test continu*
- *Pantesting*

Chaque carte propose une pratique que vous pouvez mettre en place dans votre organisation.

Chaque carte est aussi dotée d'une valeur comme dans un jeu de cartes classiques pour multiplier les usages.

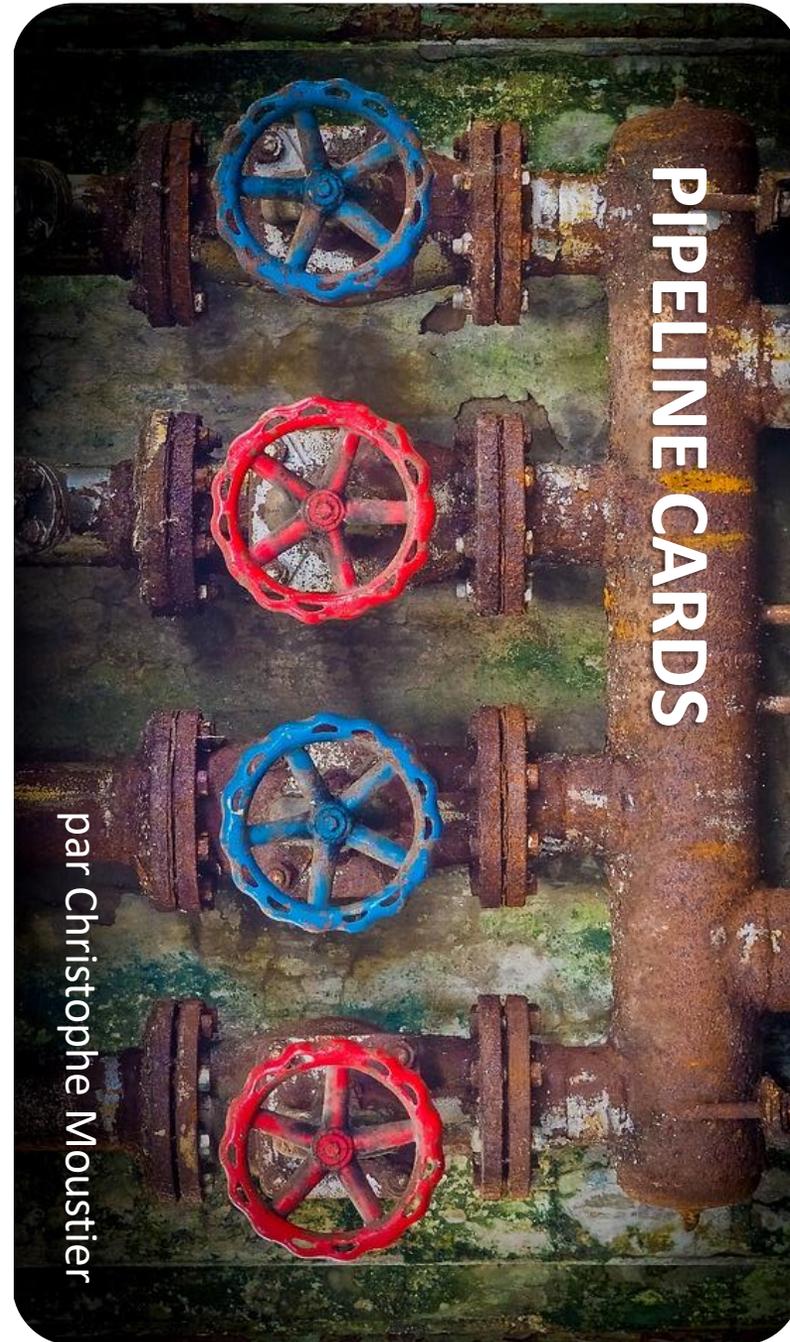


par Christophe Moustier

## Composants d'un pipeline DevOps

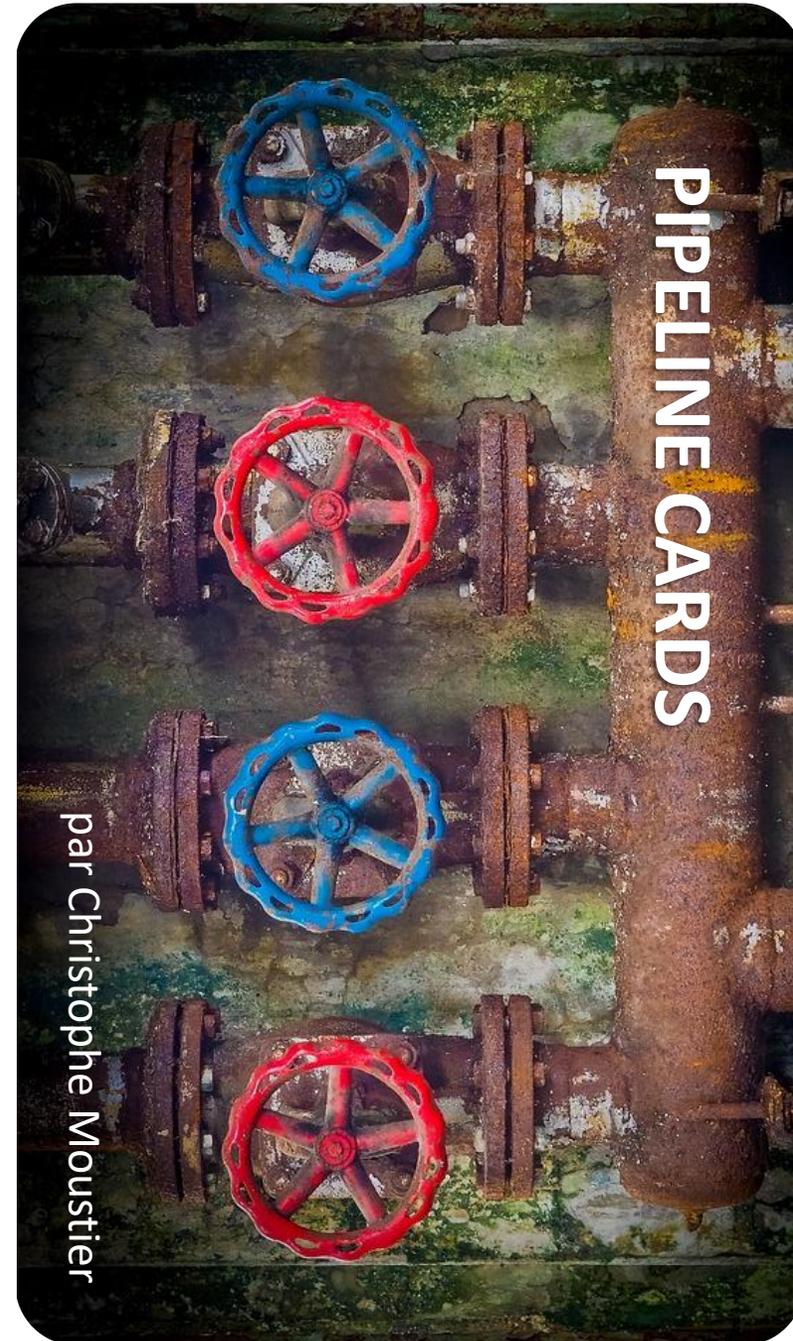
- **Continuous Integration (CI)** - intégration continue
- **Continuous Delivery** (souvent noté CD mais on utilisera "CDv" pour éviter toute ambiguïté) - distribution continue
- **Continuous Deployment** (souvent noté CD mais on utilisera "CDp" pour éviter toute ambiguïté) - déploiement continu
- **Continuous Operations (CO)** - opérations continues

### Pipeline



## Composants du Pantesting

- **Double boucle d'apprentissage** : induit un apprentissage connecté à un ordre supérieur – c'est le feedback du terrain plutôt qu'une seule conformité à des US
- **Panarchie** : chaque système suit un écocycle d'évolution avec une phase de perturbation  $\alpha$  (en provenance d'un autre écocycle), une phase d'exploration R, une phase d'accumulation K de ressources, une dernière  $\Omega$  de fin de stabilité qui relâche tout ou partie des ressources accumulées vers d'autres écocycles
- **Théorie des contraintes** : optimisation systémique des flux de production
- **Testabilité** : autant technique que sociale



par Christophe Moustier

## Références ENI.FR

« **Le Test en mode agile** »

ISBN : 978-2-409-01943-2

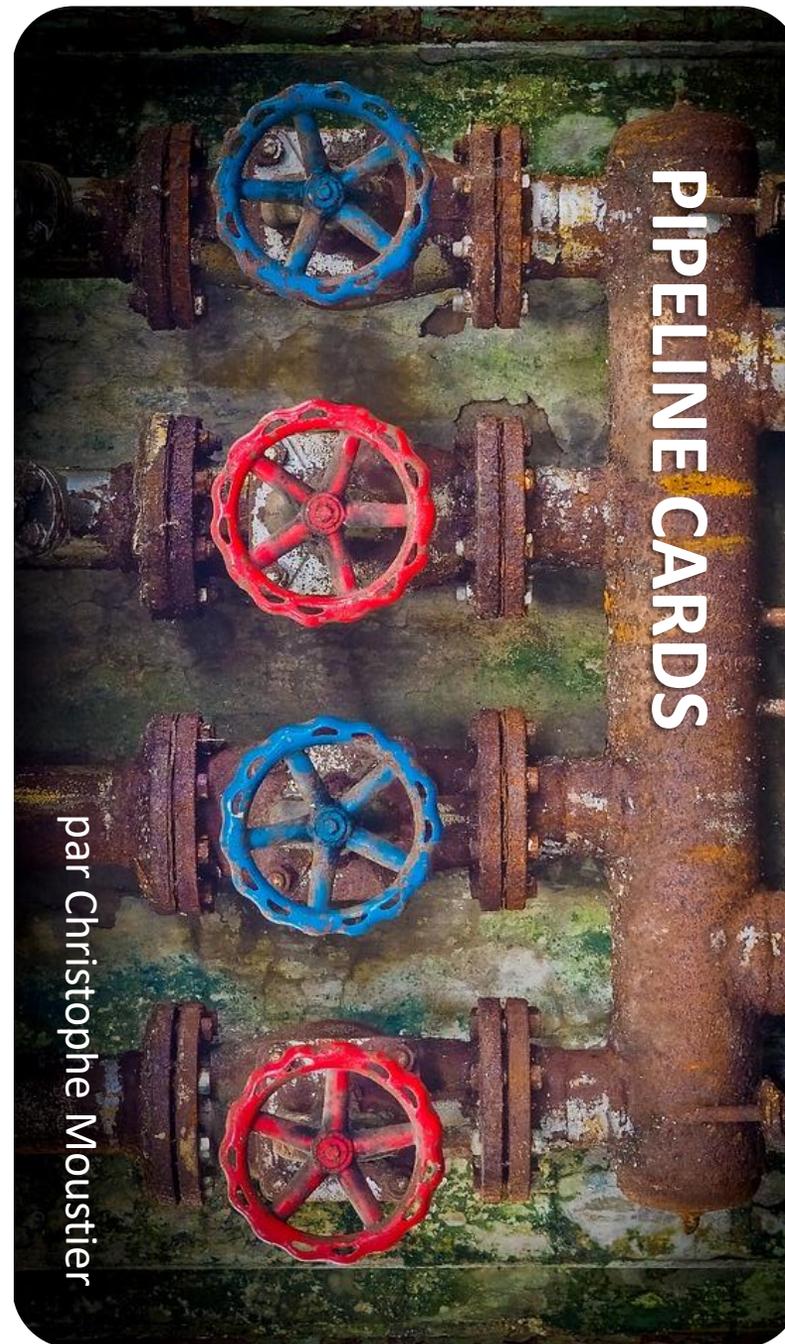
« **Conduite de tests agiles pour  
SAFe et LeSS** »

ISBN : 978-2-409-02727-7

[https://www.editions-  
eni.fr/christophe-moustier](https://www.editions-eni.fr/christophe-moustier)



[https://tinyurl.com/  
linkedin-cmoustier](https://tinyurl.com/linkedin-cmoustier)



## UTILISATIONS LUDIQUES

**#1 : Jeu des 5 familles** (2 à 5 joueurs) pour se familiariser avec chaque pratique.

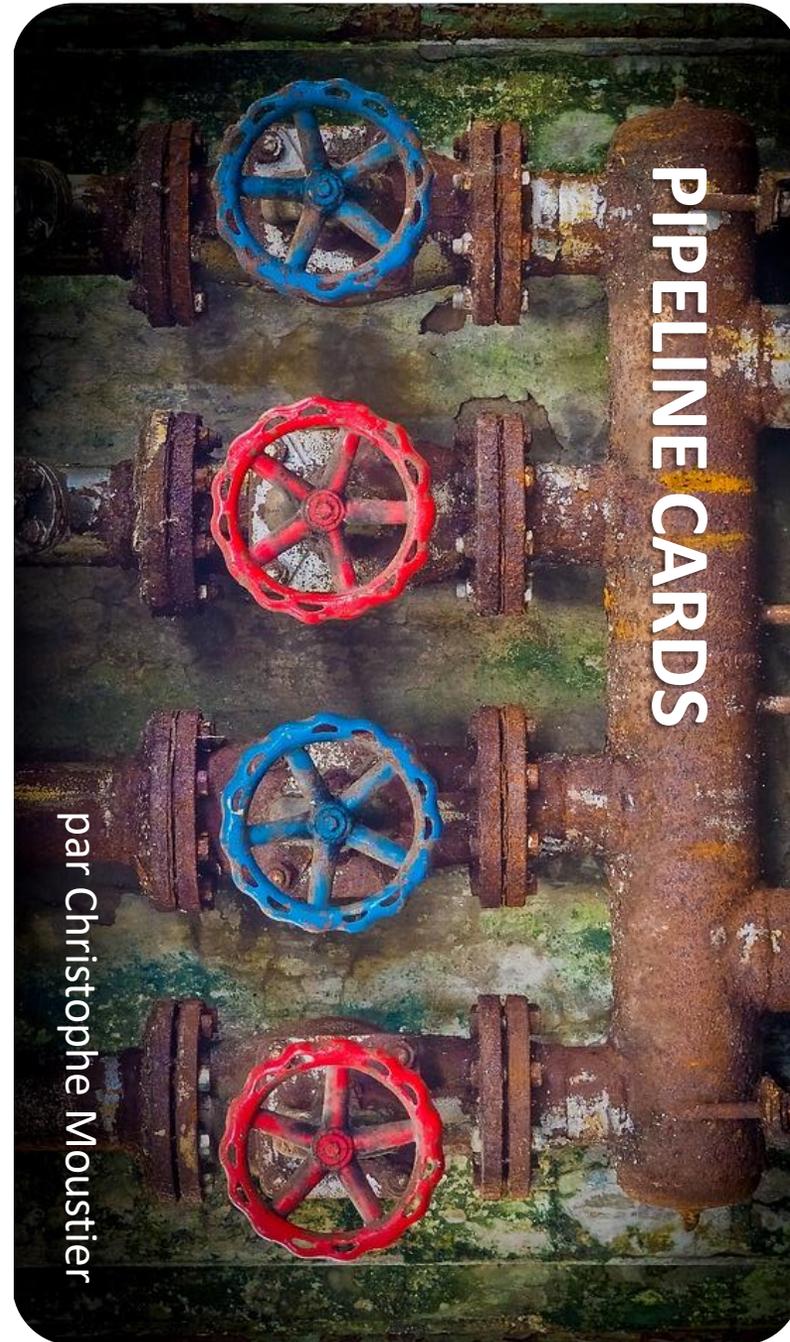
Règles : <https://www.regles-de-jeux.com/regle-des-7-familles/>.

Pour pimenter le jeu, on va demander une description de la pratique

**#2 :** C'est un jeu de cartes classique As, Roi, Dame, Valet, 10, 9, 8, 7, ... mais avec des catégories plutôt que les couleurs classiques. Vous pouvez jouer à la belote, poker, etc.

**#3 :** le jeu dispose d'un Joker (« Pipeline Power card ») pour laisser libre cours à votre imagination (voir

<https://tinyurl.com/jeuxcartes>)

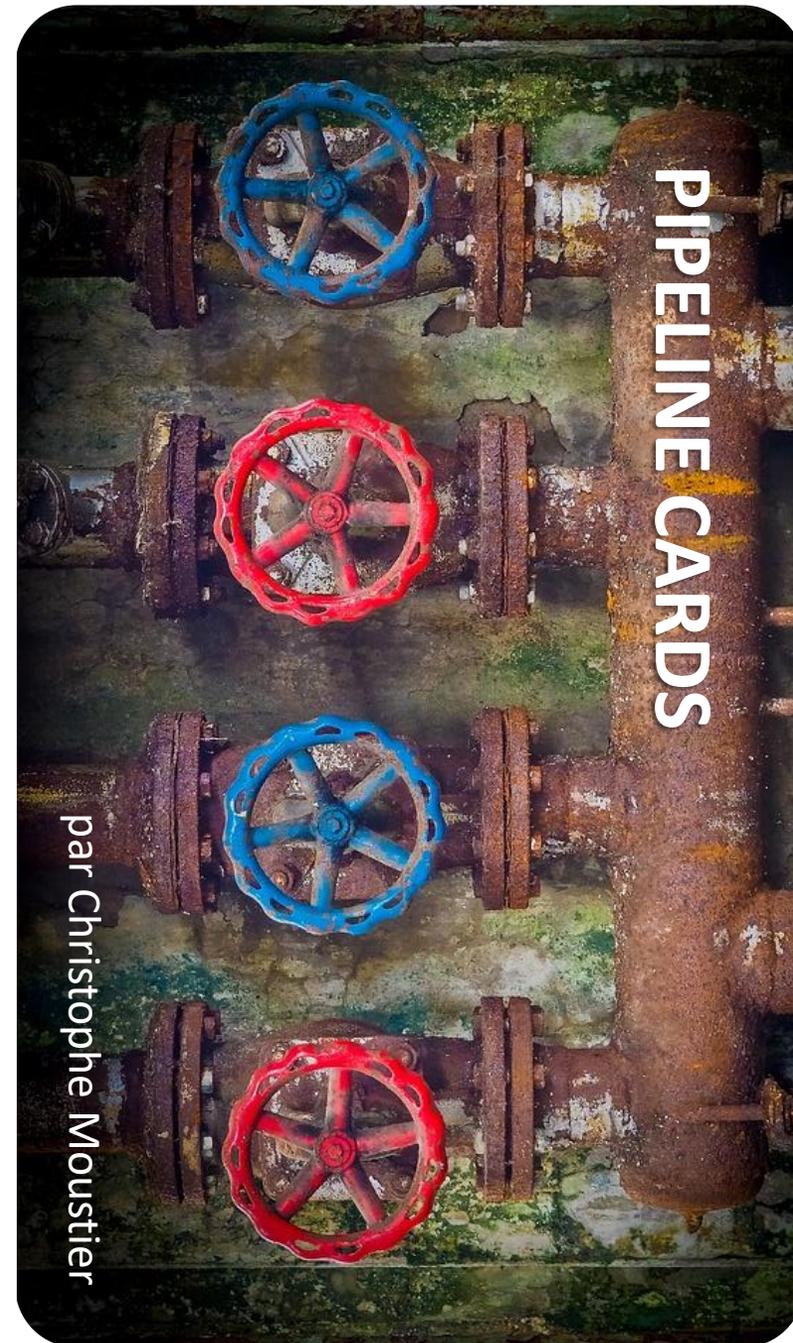


## ATELIER #1

**Evaluation collaborative des pratiques :** Préparer 5 colonnes sur une table « OK » / « Prochaine itération » / « Plus tard » / « Non applicable » / « C'est quoi ? ».

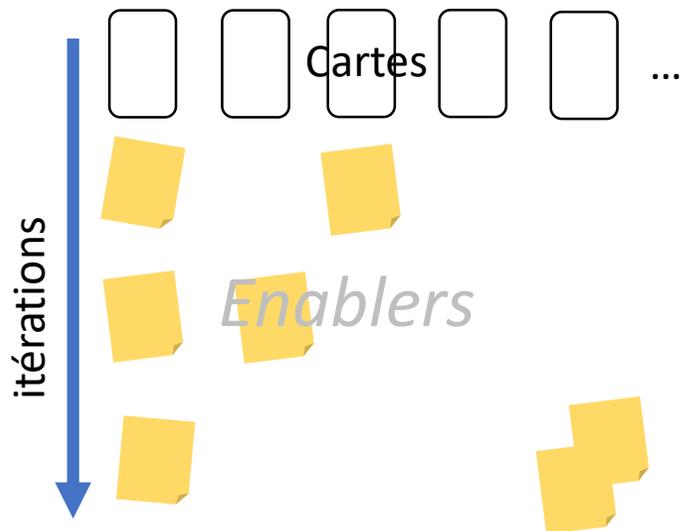
L'équipe prend chaque carte, en discute et choisit dans quelle colonne la carte doit être placée. Les deux dernières colonnes amènent à discussion pour voir comment faciliter la pratique ou acquérir plus de connaissance. Refaire l'atelier ultérieurement pour voir les progrès.

L'atelier peut être amélioré par un score sur les cartes « OK » : de 0 à 5 ( 0-Jamais / 1-Parfois / 2-Toujours / 3-Documenté/ 4-Mesuré / 5-Optimisé ou « Assis » / « Rampe » / « Marche » / « Court » / « Vole »)



## ATELIER #2

**Pipeline mapping :** Pour les cartes qui sont retenues, imaginer avec l'équipe les Enablers qui vont faire progresser la pratique vers un niveau plus élevé





BUILD



## Construction de la solution

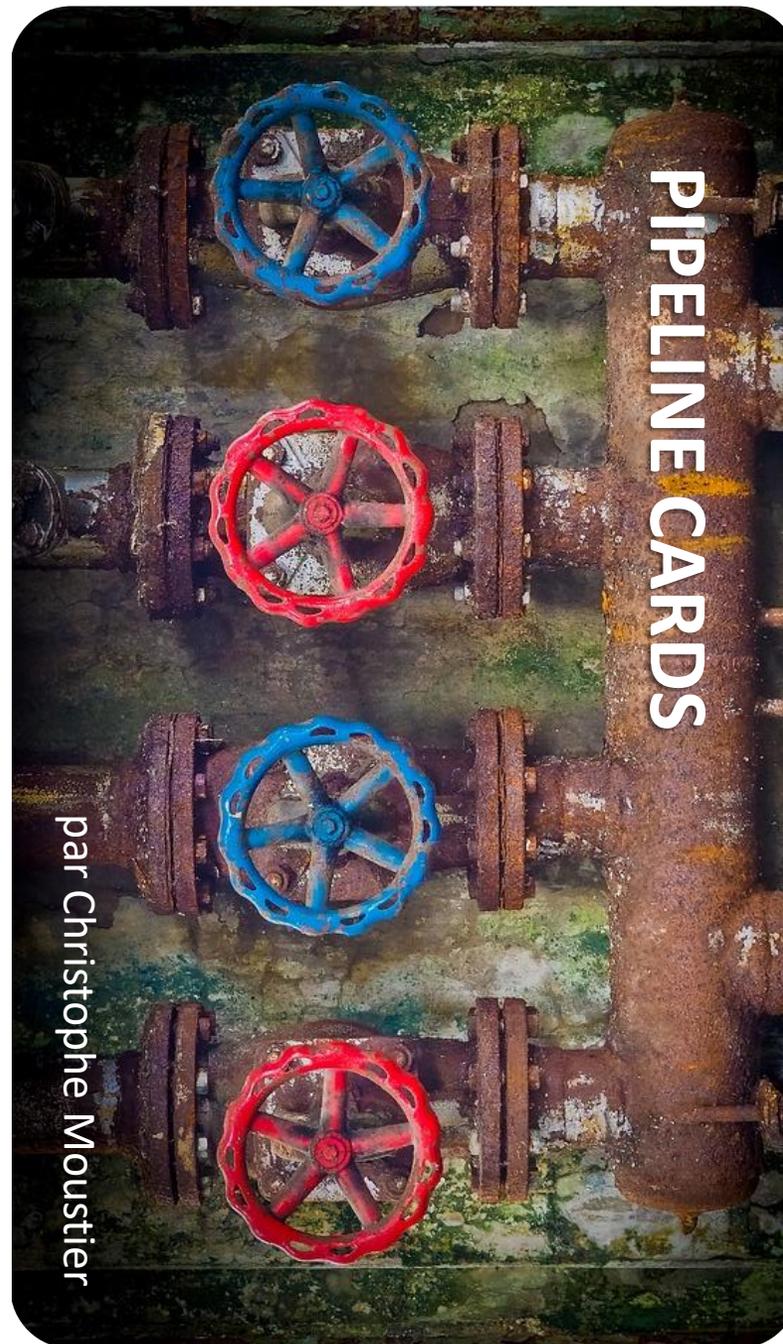
Cette activité de base consiste à compiler le code et générer la base de données avec une configuration d'usine

CI

CDv

CDp

CO





BUILD



## Préparation des paquetages en fonction des plateformes

Chaque environnement  
cible doit pouvoir  
recevoir ses contenus  
spécialement configurés  
pour être prêt à déployer

CI

CDv

CDp

CO





BUILD



## Déploiement itératif et incrémental

En plus du déploiement  
des incréments à chaque  
itération, on pourra  
utiliser des techniques de  
Canary Releasing et Dark  
Launch pour déployer à  
tous moments

CI

CDv

CDp

CO





BUILD



## Développement agile

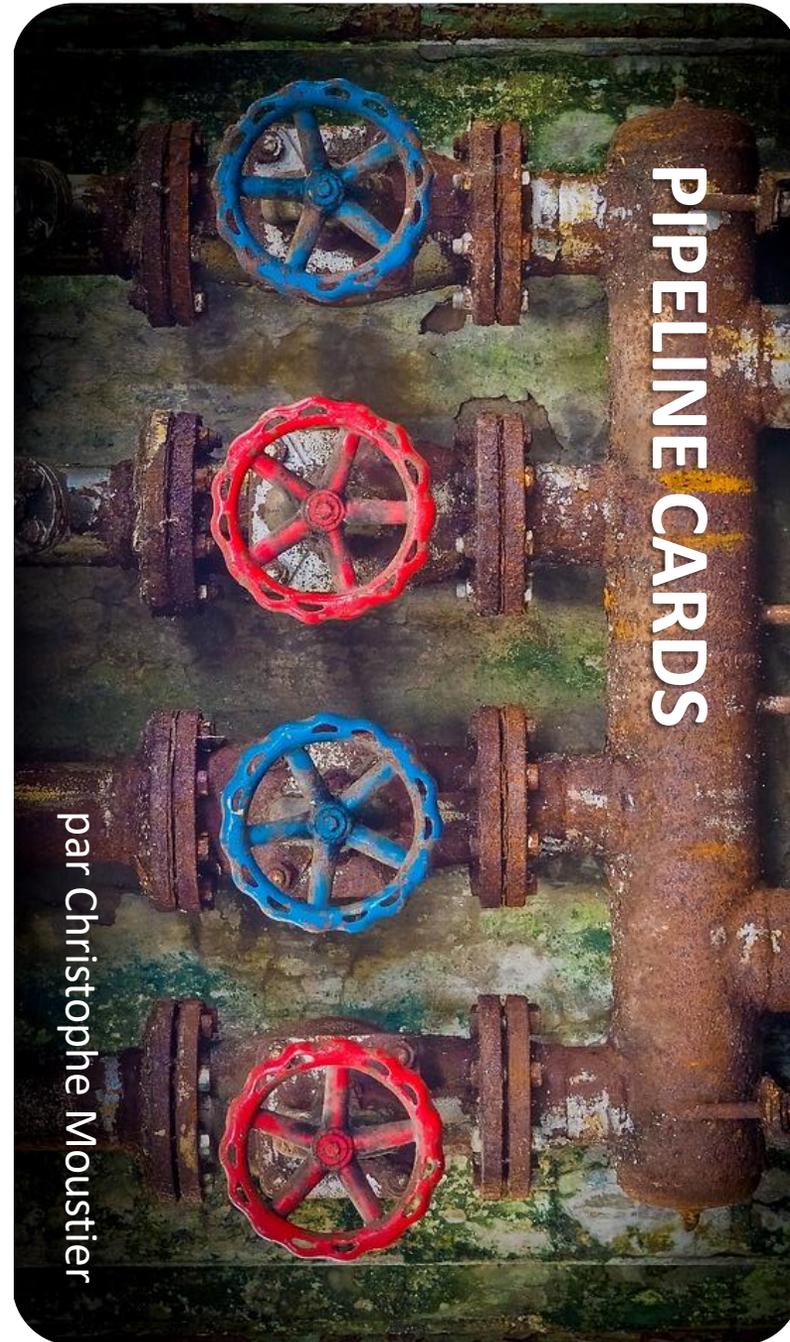
Les valeurs et principes  
de l'agile sont adoptés

CI

CDv

CDp

CO





BUILD

Equilibrer la charge de  
développement et les  
impératifs  
d'exploitation

L'approche SRE propose  
la moitié du temps pour  
chaque type d'activité

CI

CDv

CDp

CO





BUILD

## Génération de la documentation

Dans l'état d'esprit "*Living Documentation*", le code contient de la documentation sur la solution

CI

CDv

CDp

CO





BUILD

IaC

"Infrastructure as Code" -  
la génération de  
l'infrastructure est  
réalisée à l'aide de code  
ou de fichiers de  
configuration

CI

CDv

CDp

CO





## BUILD

### Intégration fréquente du code

Le code de chaque développeur est intégré au moins une fois par jour. Plus la fréquence est élevée, moins il y a de risque de casser le code. Cette fréquence peut ainsi se faire régulièrement à heure fixe ou à chaque soumission de code

CI

CDv

CDp

CO





## BUILD

### Pas de branche de développement

On développe tous sur le "tronc" pour ne pas avoir de longue phase d'intégration de code avec les autres développeurs  
(lien avec « Intégration fréquente du code »)

CI

CDv

CDp

CO





BUILD

## Traitement rapide du pipeline

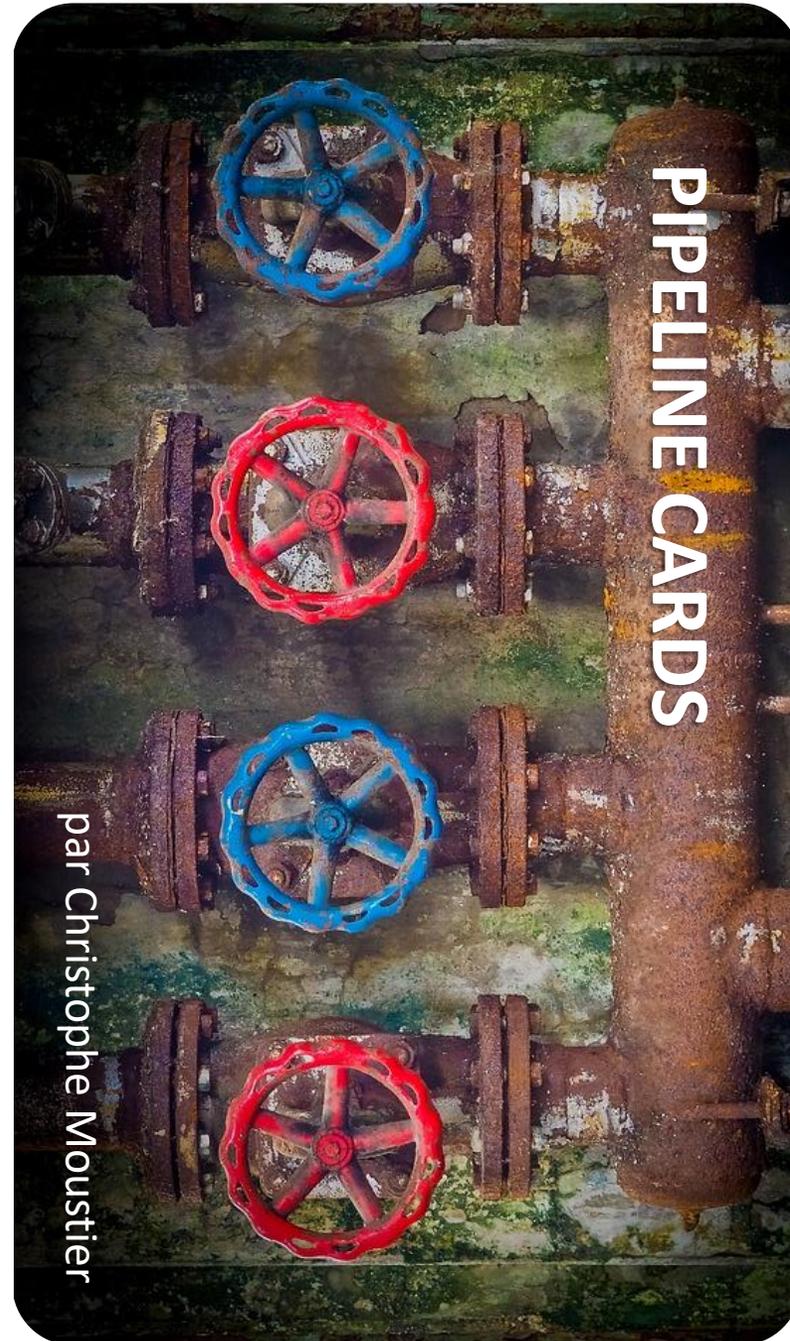
Pour faciliter la soumission de code et éviter une accumulation de changements qui seraient alors trop importants, il est crucial que le temps de traitement du pipeline soit court, c'est-à-dire moins de 10 minutes

CI

CDv

CDp

CO





BUILD

## Code toujours prêt

L'équipe doit s'assurer en priorité que le code est toujours prêt à être poussé dans le pipeline

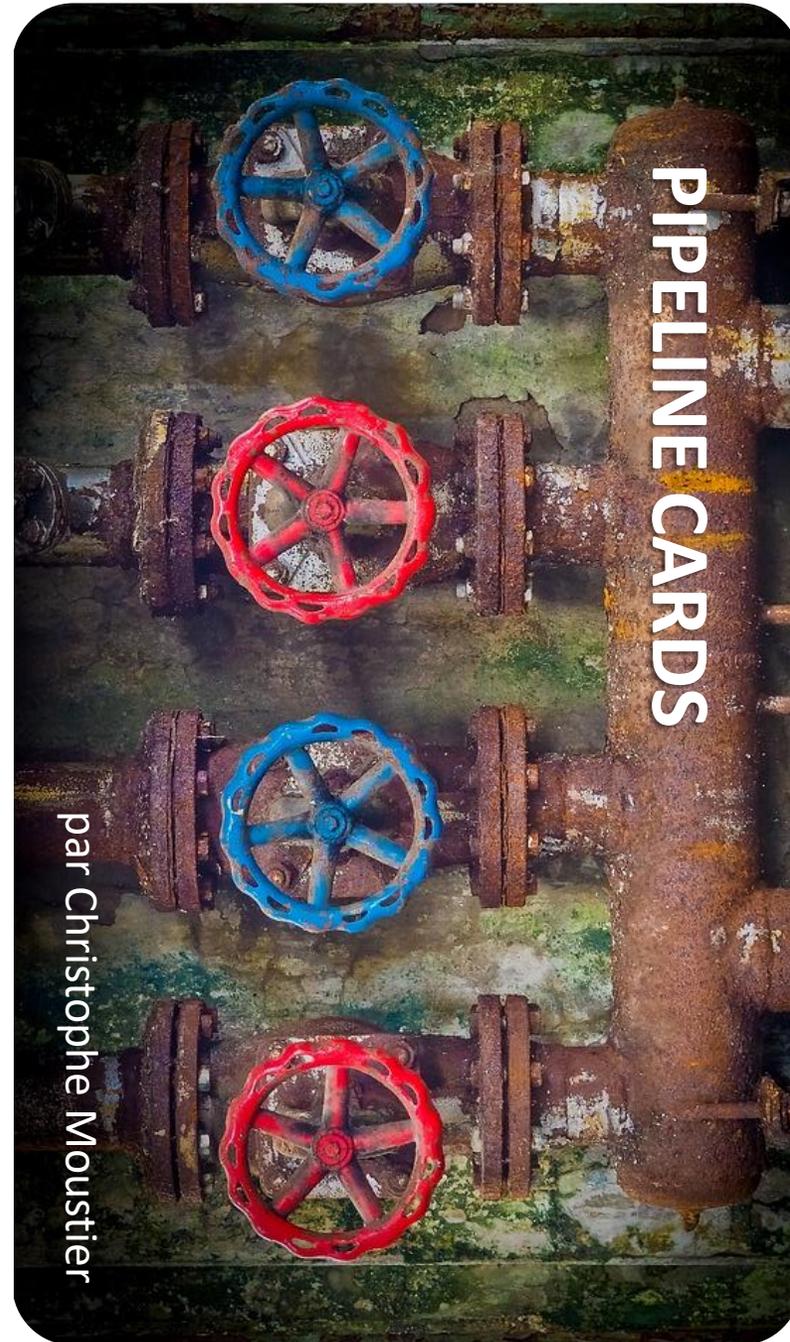
4

CI

CDv

CDp

CO





BUILD

## Dark Launch

Le nouveau code est déployé en production sans impacter l'utilisation de la version en exploitation, notamment grâce à des Feature Flags

CI

CDv

CDp

CO





BUILD

## Refactoring régulier

L'ancien code est régulièrement réusiné sans régression (ex. par du TDD) pour résorber la dette technique

CI

CDv

CDp

CO





ENVIRONNEMENT



## Déploiement sur les environnements cibles

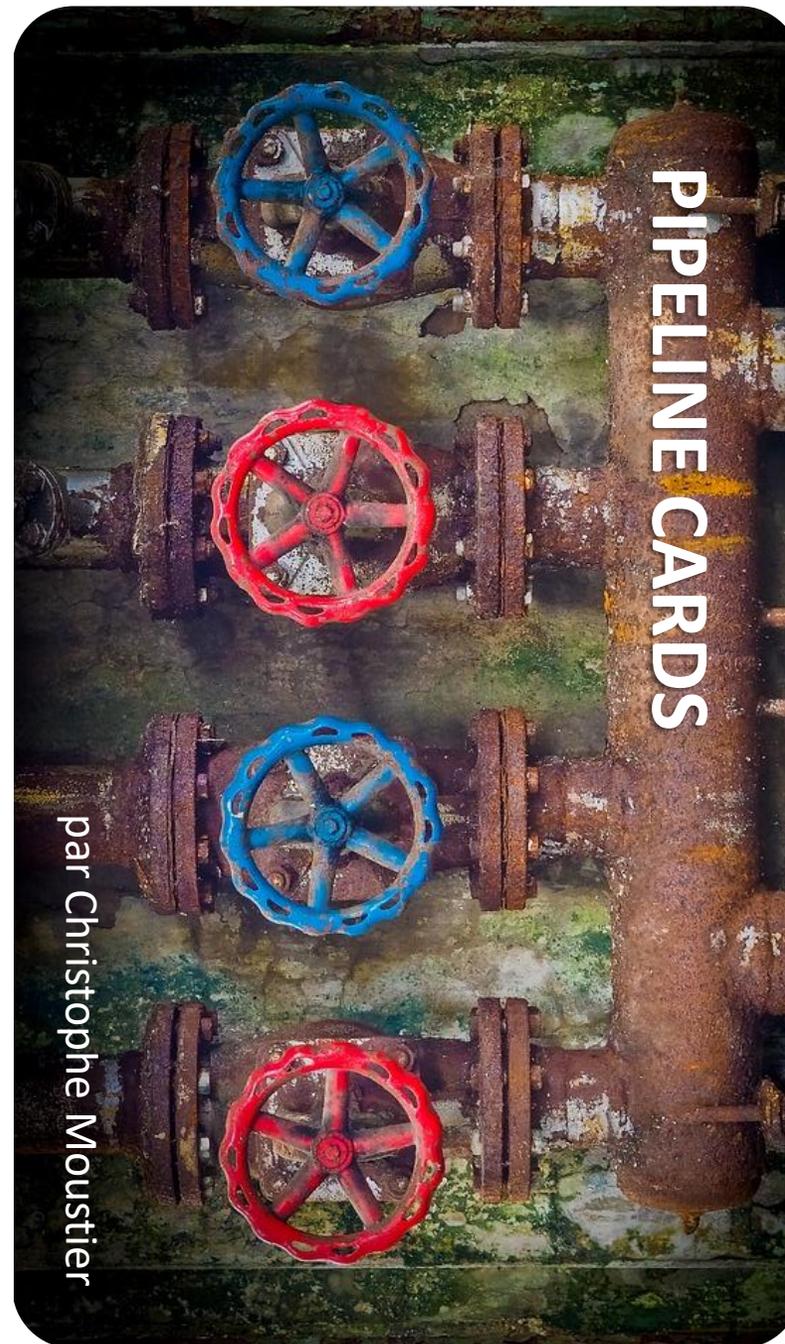
Chaque partie de la solution avec sa configuration est déployée sur l'environnement qui lui est destiné

CI

CDv

**CDp**

CO





## ENVIRONNEMENT

### Pipeline structuré

Le pipeline doit avoir des étapes bien distinctes d'assemblage, publication et exécution afin de maximiser la capacité de déploiement et rassurer les équipes méfiantes

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Environnements clonés

Les environnements de travail et de test sont proches de celui de production

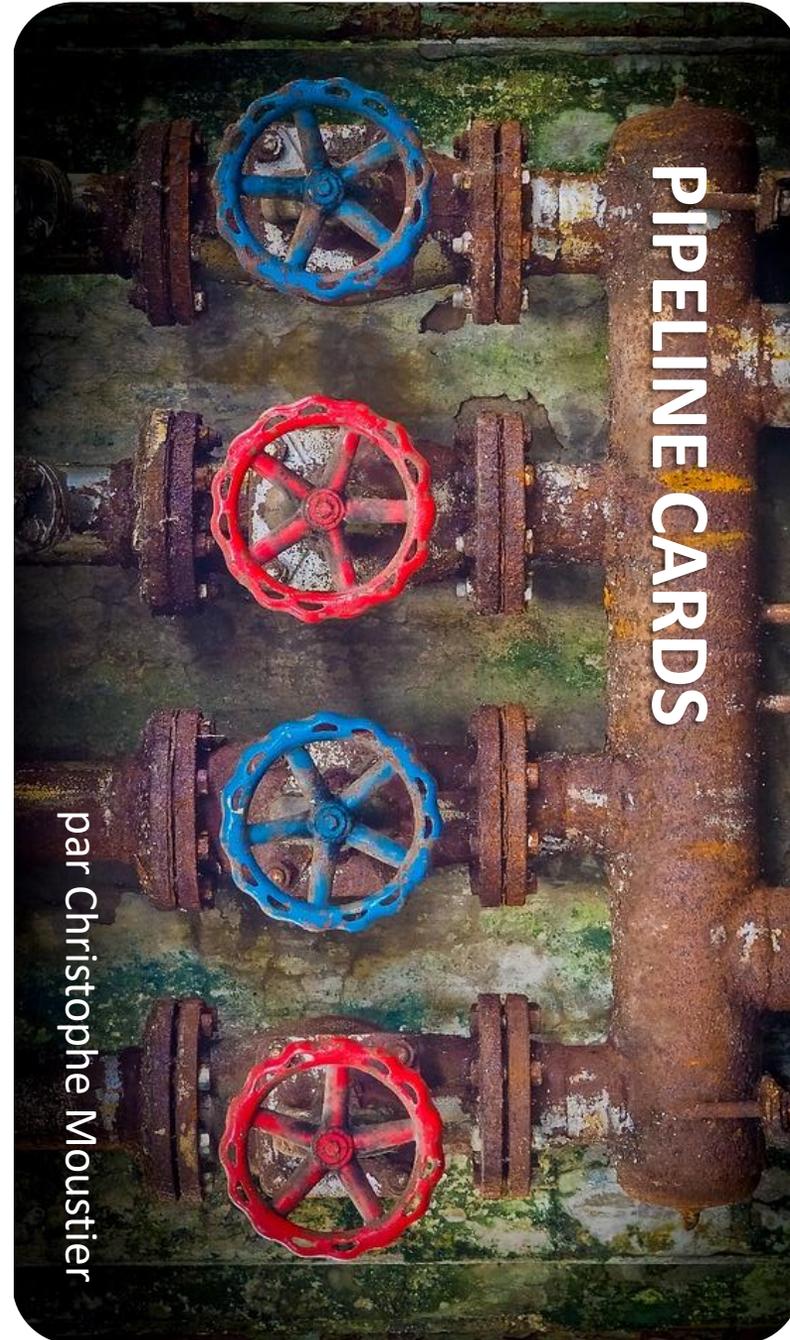
D

CI

CDv

CDp

CO



par Christophe Moustier



## ENVIRONNEMENT

### Equipes "2 Pizzas"

Les équipes sont de  
petite taille (on peut les  
nourrir avec 2 pizzas)

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Equipes de fonctionnalités

De telles équipes sont capables d'intervenir sur pratiquement toutes les parties de la solution et prennent en charge quelques fonctionnalités sur tous les aspects techniques

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Equipes en mode start-up

Les équipes ont pleine  
autonomie sur leur  
périmètre fonctionnel

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Gestion de la capacité de service

Par la planification des évolutions fonctionnelles, en réponse à la demande du marché ou pour une certaine tolérance aux pannes, les serveurs seront générés ou dissous en fonction des circonstances

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Infrastructure scalable

Pour que l'infrastructure puisse évoluer avec l'application, son architecture doit être modulaire, redondée et administrée automatiquement ; par ailleurs, le code ne doit prendre pour hypothèse l'identification du serveur sur lequel il est exécuté ni sa pérennité

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Participation des développeurs aux tâches d'exploitation

Cela participe à l'acculturation des équipes de développement et aide à une automatisation durable des opérations manuelles - voir la carte "Equipes X-Teams"

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Pipeline scalable

Faciliter un traitement rapide du pipeline lorsque les équipes sont nombreuses, ex.

- Améliorer les temps d'exécution
- Proposer une chaîne d'intégration continue à plusieurs étages
- Au niveau de la fréquence (au commit ou à heures fixes)

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Processus scalable

Chaque appel du service doit s'appuyer sur un processus différent à chaque fois afin de permettre la mise à l'échelle des appels des services et de ne pas impacter tous les autres appels en cas de plantage d'un des appels (ex. gestion des sessions)

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Pipelines secondaires

En plus du pipeline destiné à traiter le code qui part rapidement et directement en exploitation, d'autres pipelines sont lancés à heure fixe ou le weekend sur des traitements trop longs

CI

CDv

CDp

CO





## ENVIRONNEMENT

### Zero Downtime Deployment

Montée de la version N  
vers la N+1 sans  
interruption du service  
entre chaque mise en  
production  
(ex. Blue/Green  
Deployment)

CI

CDv

CDp

CO





SOLUTION



## Politique de traitement des anomalies

Degrés de traitements :

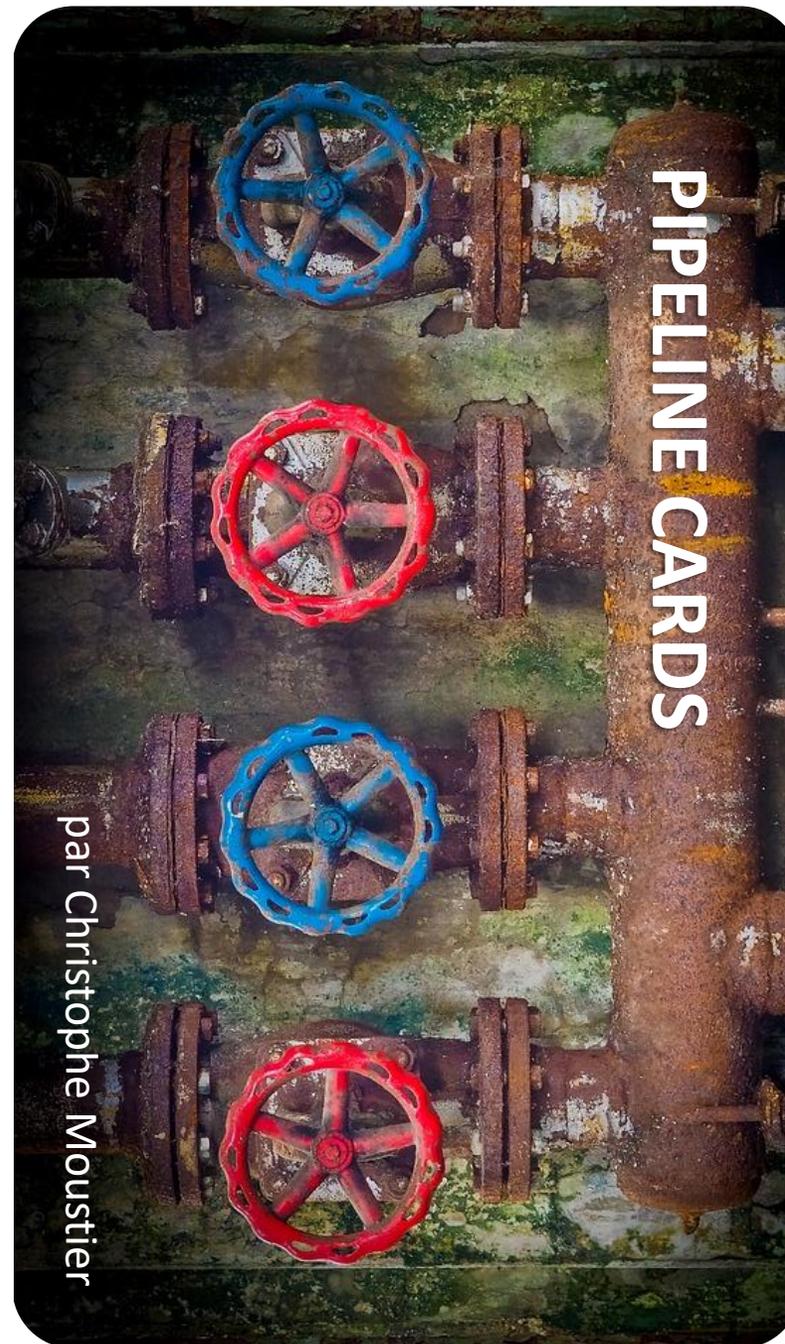
- Le signal d'anomalie est supprimé ou stoppé
- Il est encapsulé et pas propagé
- Il a une portée limitée
- Il est traité isolément
- Il est traité en tous lieux
- Tentative de RCA

CI

CDv

CDp

CO





## SOLUTION

### Accès réseau

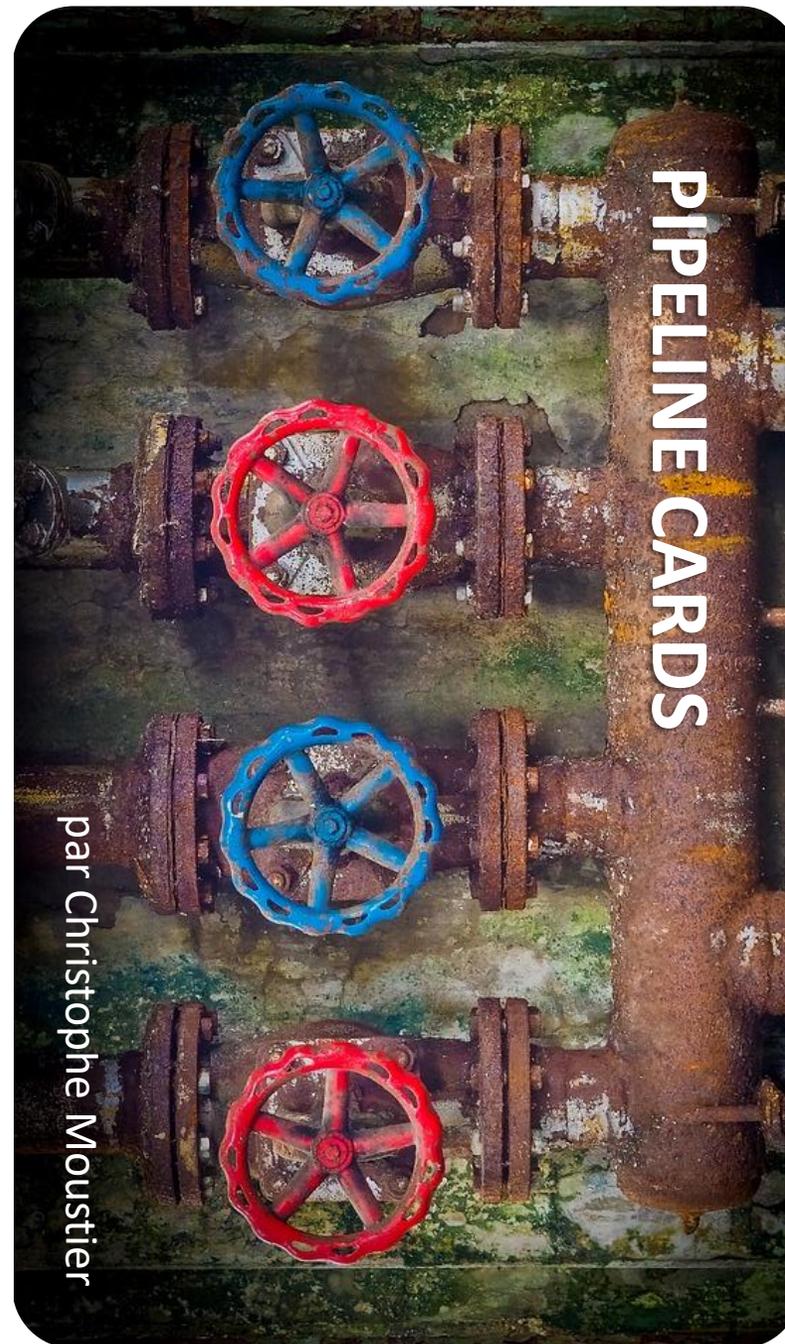
Les services sont rendus disponibles aux autres par un port dont la configuration peut être modifiée en production, notamment pour gérer efficacement le routage

CI

CDv

CDp

CO





## SOLUTION

### Administration dans la configuration

Les tâches d'administration sont conservées en gestion de configuration et packagées avec le service

CI

CDv

CDp

CO





## SOLUTION

### Arrêt inopiné

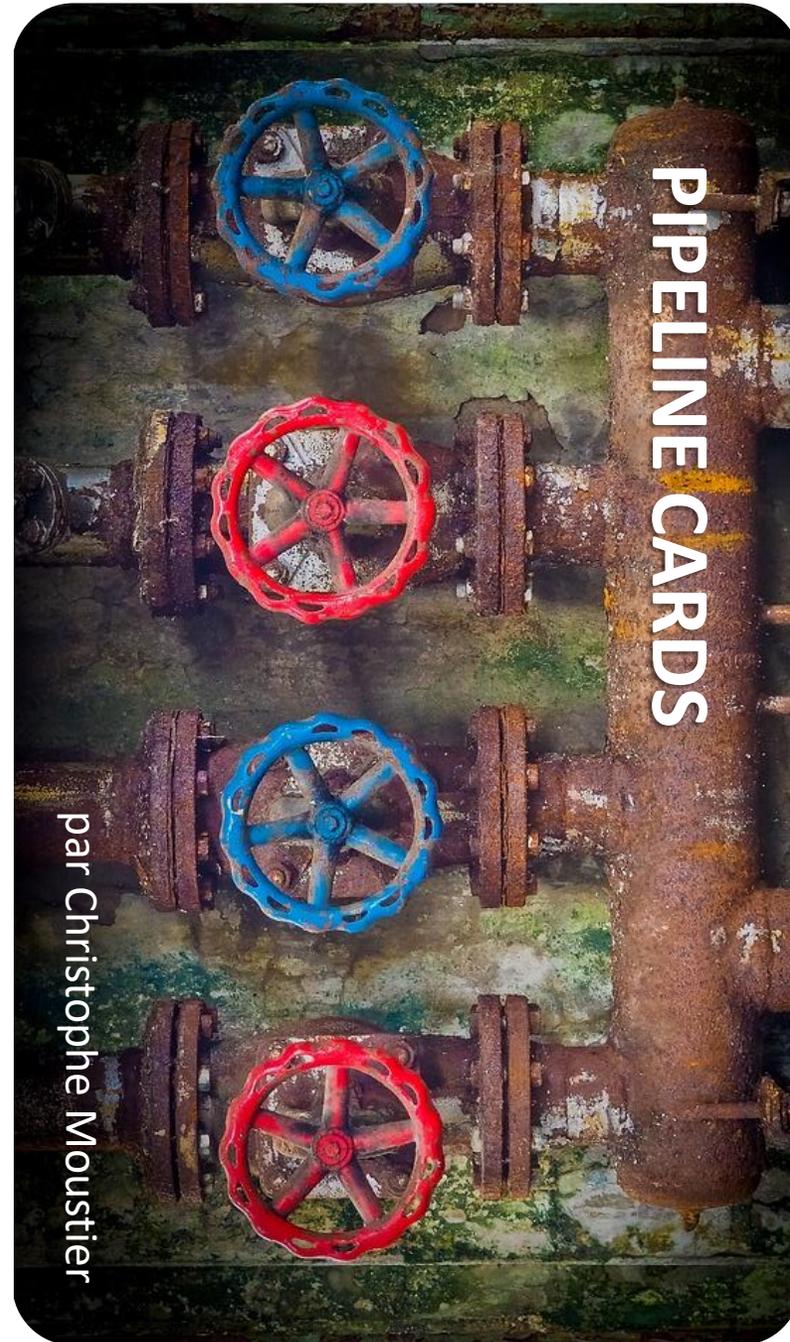
Les services sont  
démarrés ou arrêtés  
rapidement à n'importe  
quel instant

CI

CDv

CDp

CO





## SOLUTION

### Aucune dépendance implicite

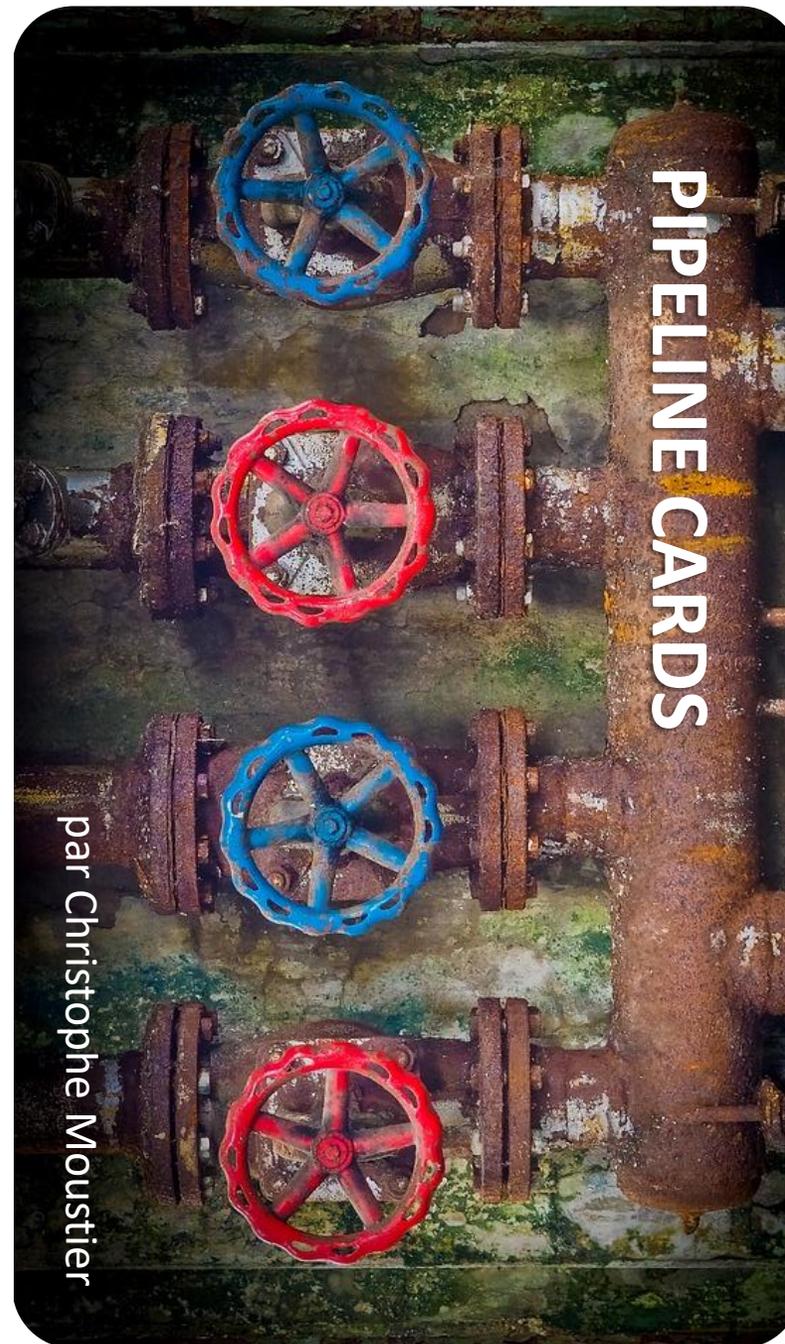
Toutes les dépendances  
sont déclarées – aucune  
dépendance implicite  
d'un système, d'un outil  
ou d'une librairie

CI

CDv

CDp

CO





## SOLUTION

### Configuration dans l'environnement

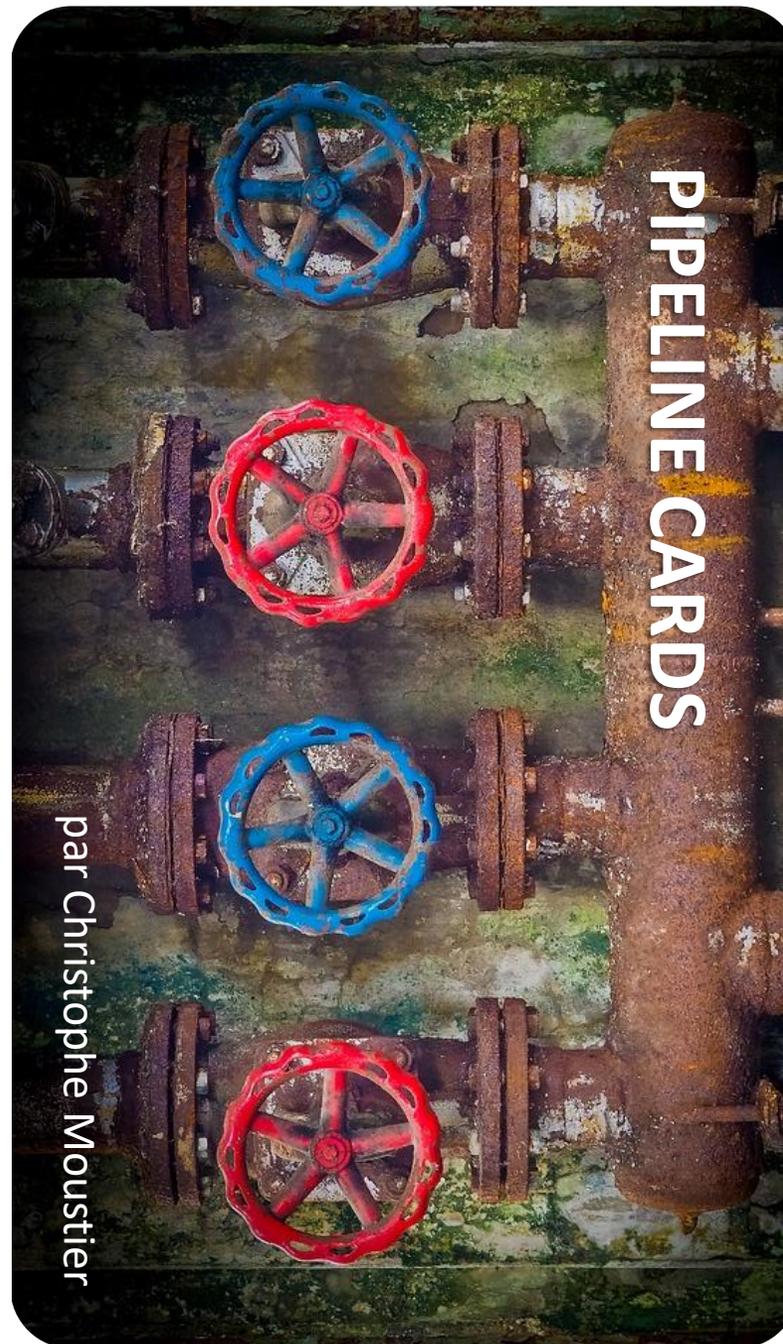
Les variations de configuration entre les environnements sont stockées dans l'environnement et non dans le code

CI

CDv

CDp

CO





## SOLUTION

### Feature Toggling

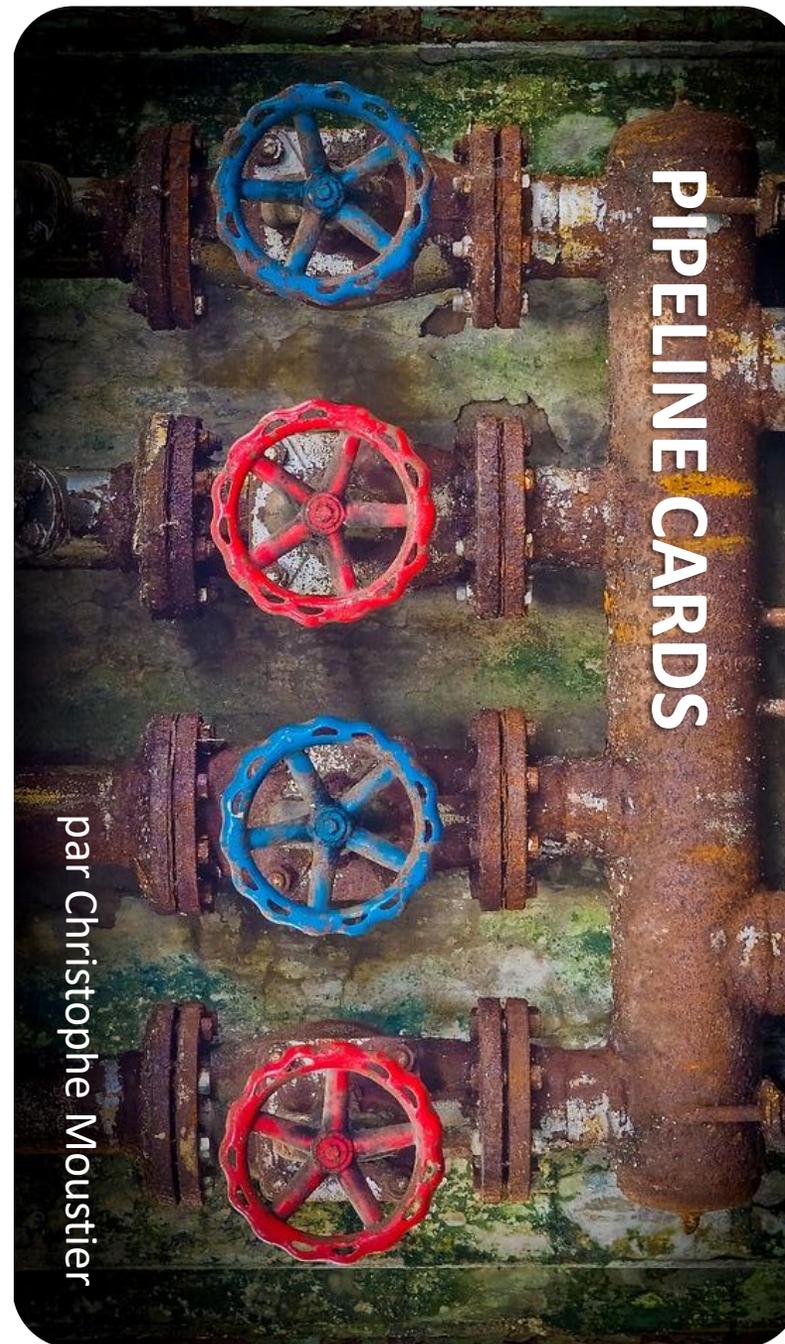
Dissociation du déploiement et de la mise à disposition (releasing) pour activations ciblées de fonctionnalités à la demande

CI

CDv

CDp

CO





## SOLUTION

### Interfaces et services

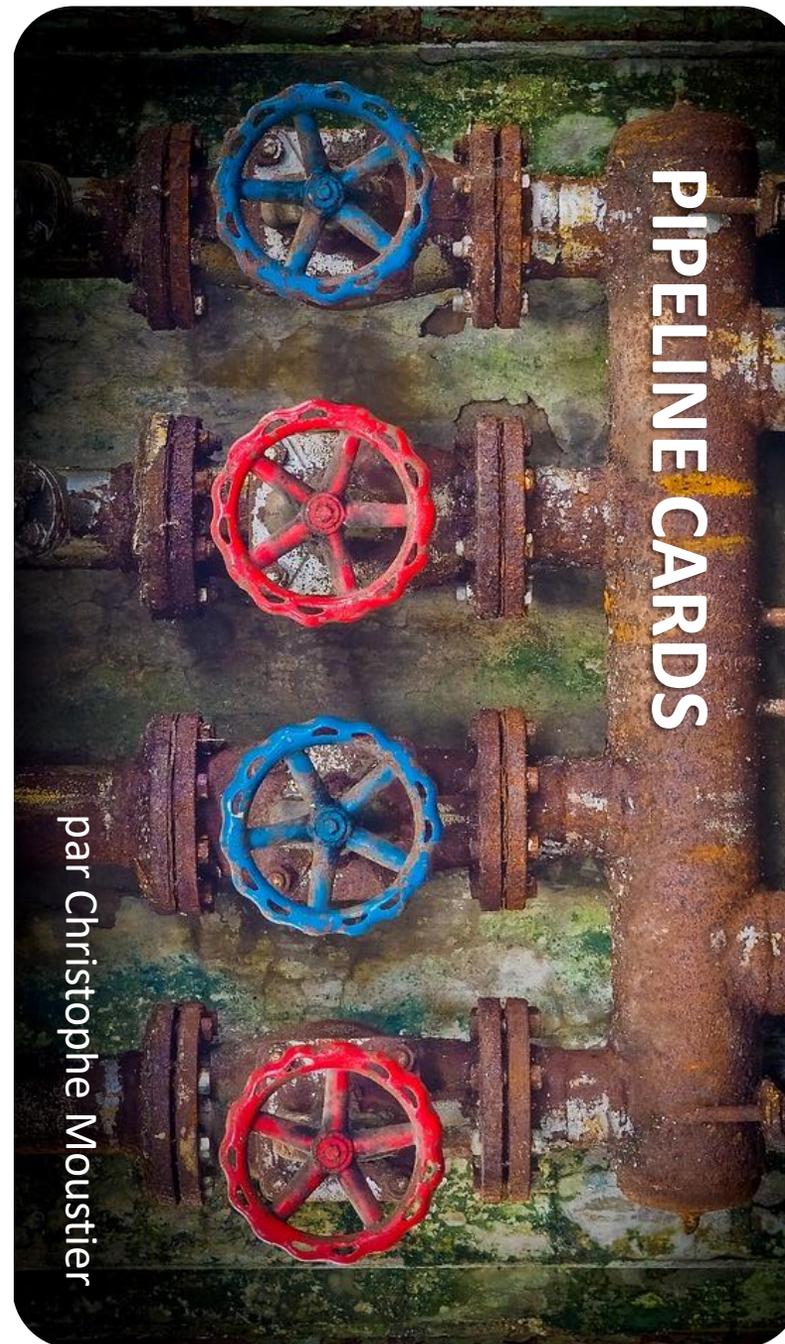
Le SI est constitué de services actionnables par une interface plutôt qu'une application monolithique

CI

CDv

CDp

CO





## SOLUTION

### Logs externes

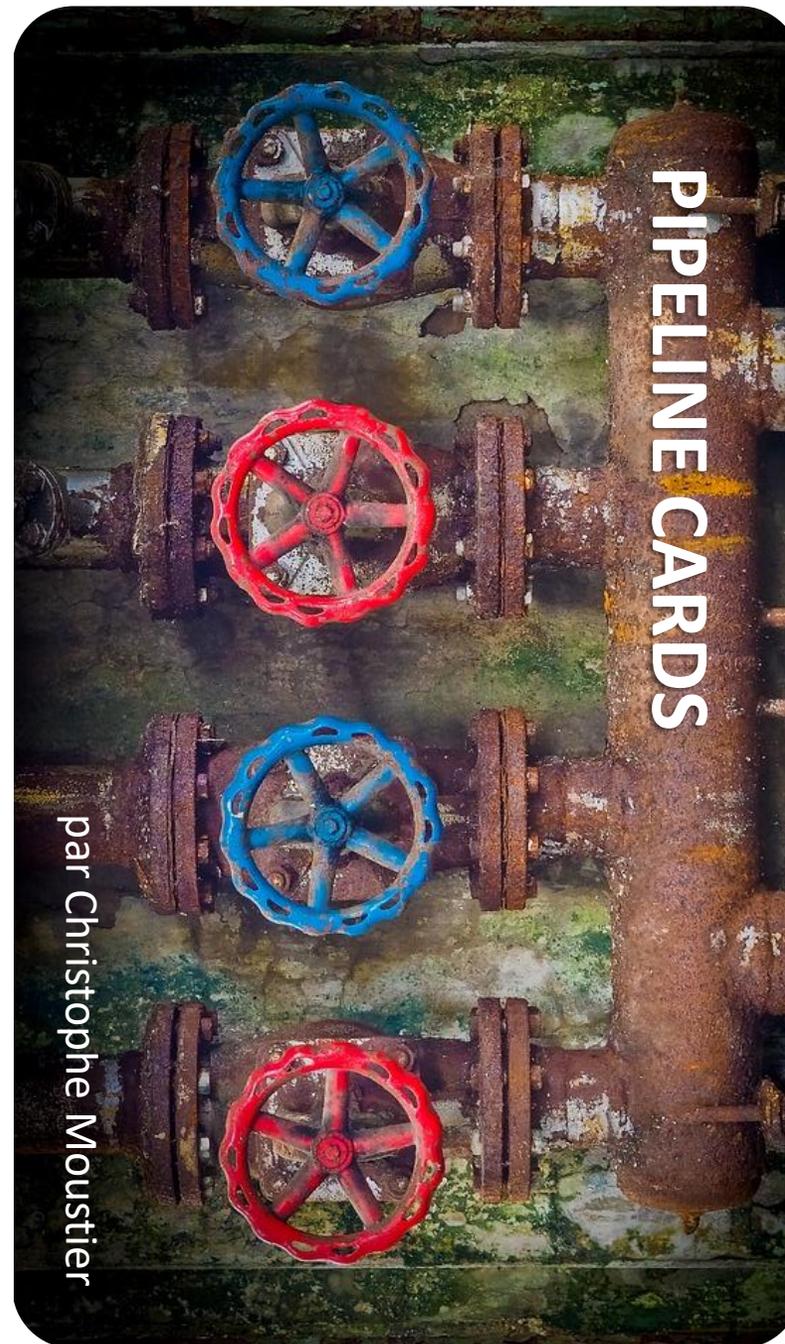
Les services doivent déléguer la production de logs à des services externes – les logs sont ainsi traités comme des flux dont le stockage est délégué à des services tiers spécifiques

CI

CDv

CDp

CO





SOLUTION

## Microservices

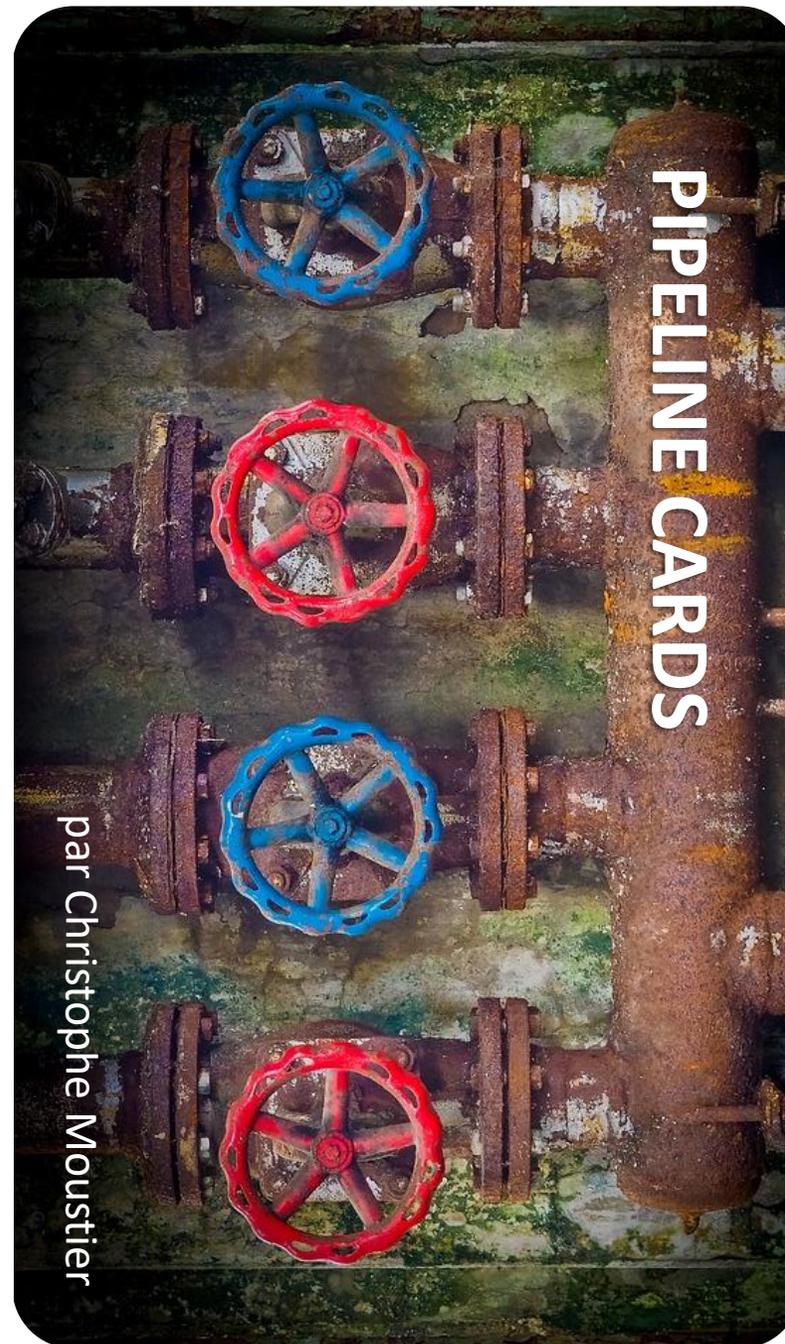
Le plus gros monolithe valide, c'est-à-dire suffisamment petit pour être modifiable de façon atomique et suffisamment gros pour ne pas augmenter la complexité du système

CI

CDv

CDp

CO





## SOLUTION

### Services externes en configuration

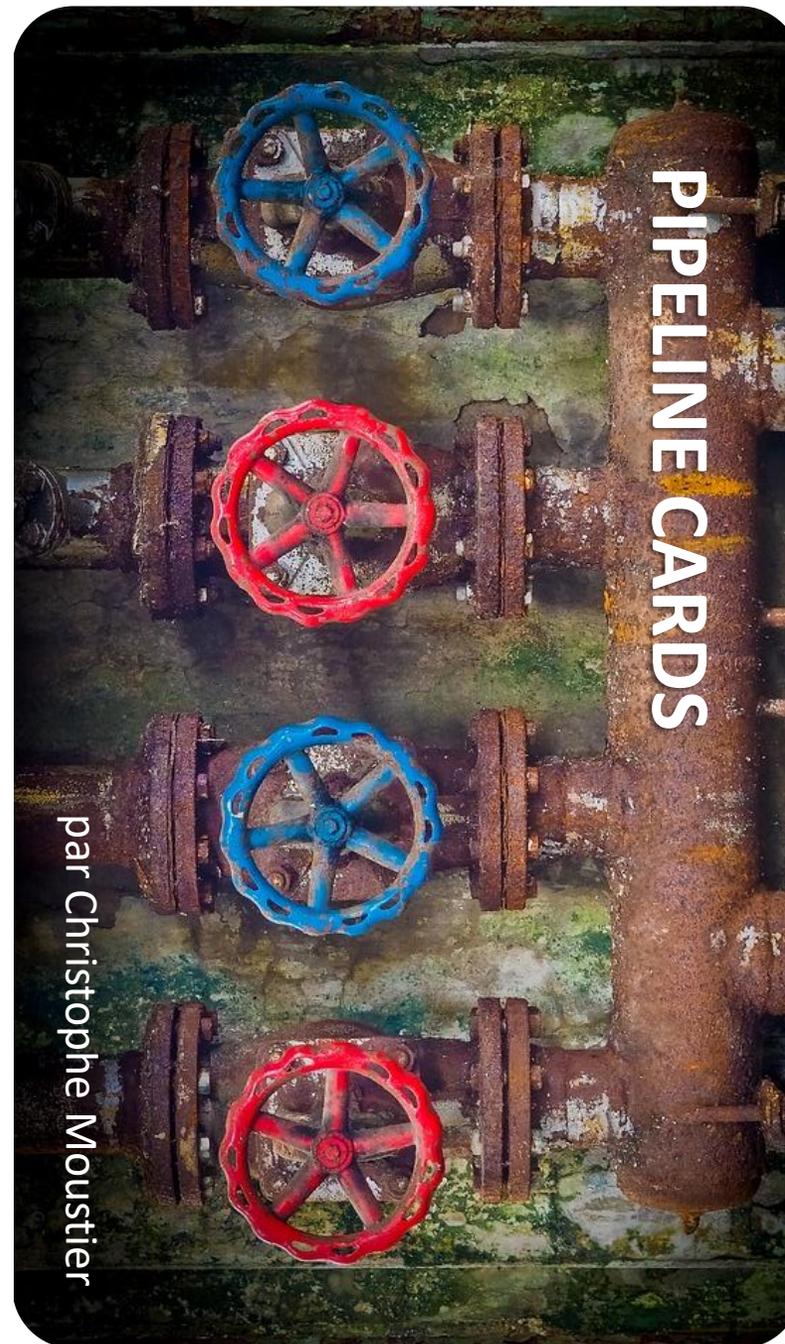
Les services externes doivent pouvoir être attachés ou détachés en configurant l'environnement sans avoir à toucher le code

CI

CDv

CDp

CO





## SOLUTION

### Services immuables

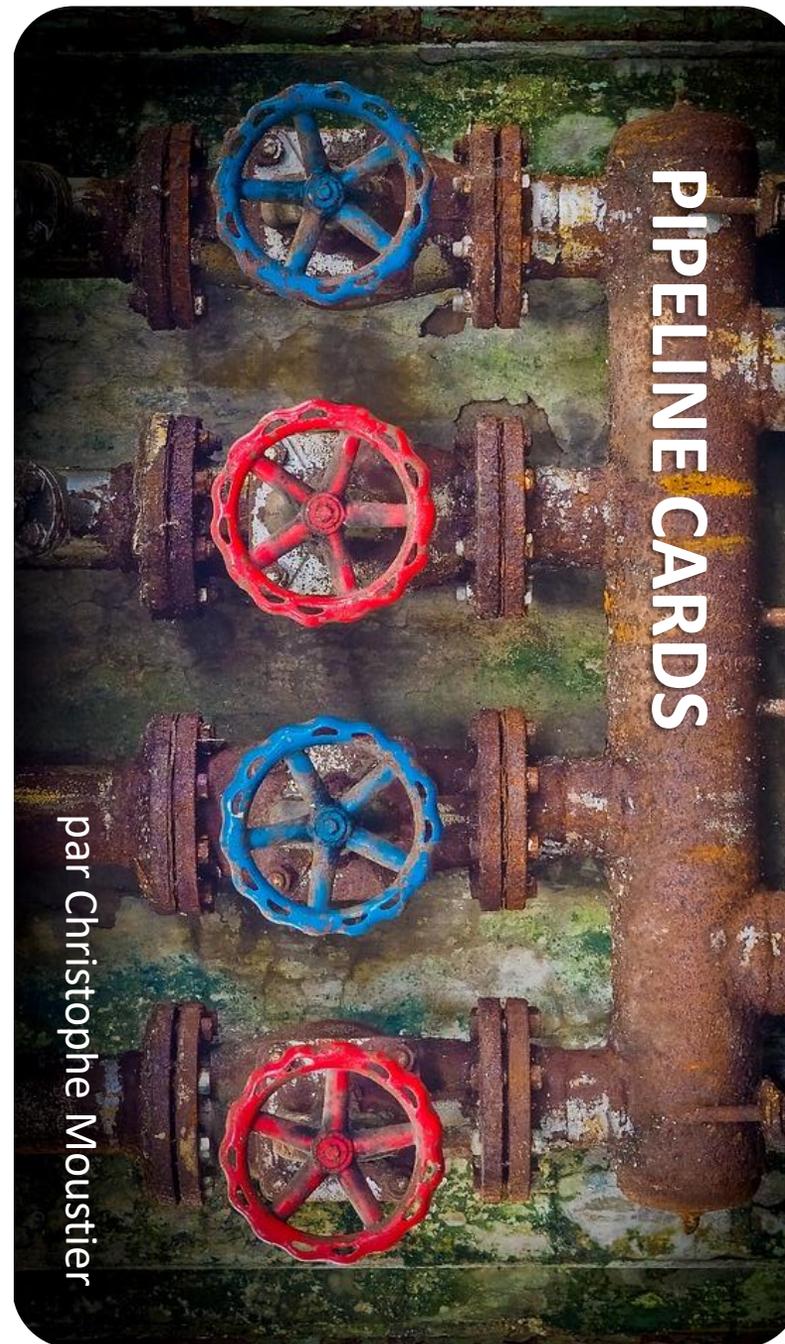
Une seule base de code pour un service afin de disposer de releases immuables qui peuvent être déployées en différents environnements

CI

CDv

CDp

CO





## SOLUTION

### Services sans état

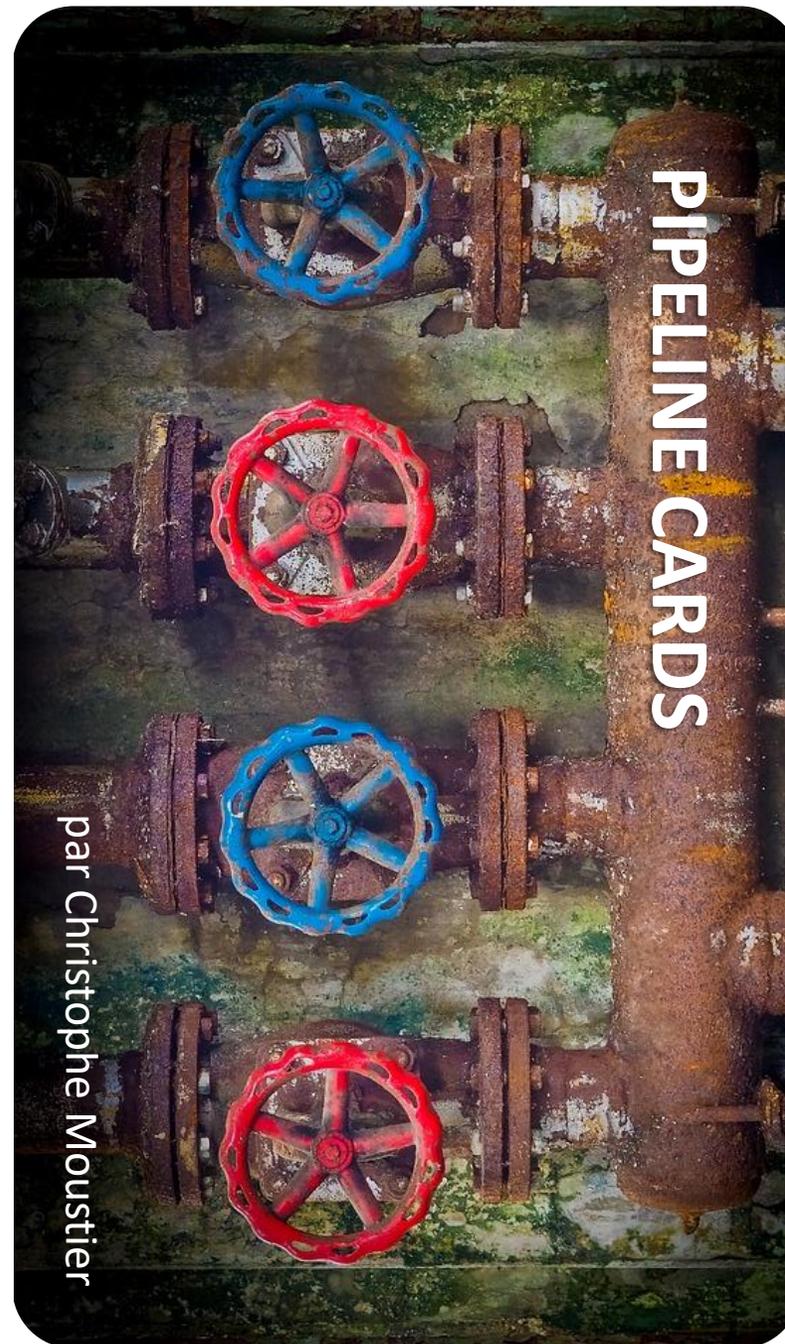
Les services devraient être conçus « sans état »  
- l'état en cours du processus devrait être stocké dans un service tiers afin de permettre de reprendre un processus en cas de panne

CI

CDv

CDp

CO





TEST CONTINU



**Assurer un feedback  
sur la bonne santé des  
ressources, du métier  
et du business**

Des métriques sur le SI et  
ses environnements  
permettent de donner un  
tableau de bord  
nécessaire à la détection  
au plus tôt (alertes,  
tickets, logs)

CI

CDv

CDp

CO





## TEST CONTINU

### Afficher l'état du pipeline

De façon bien visible, indiquer l'état du pipeline avec l'étape bloquée constitue un excellent Andon et mobilise les équipes sur la bonne santé du processus de déploiement

CI

CDv

CDp

CO





## TEST CONTINU

### Budget d'erreurs

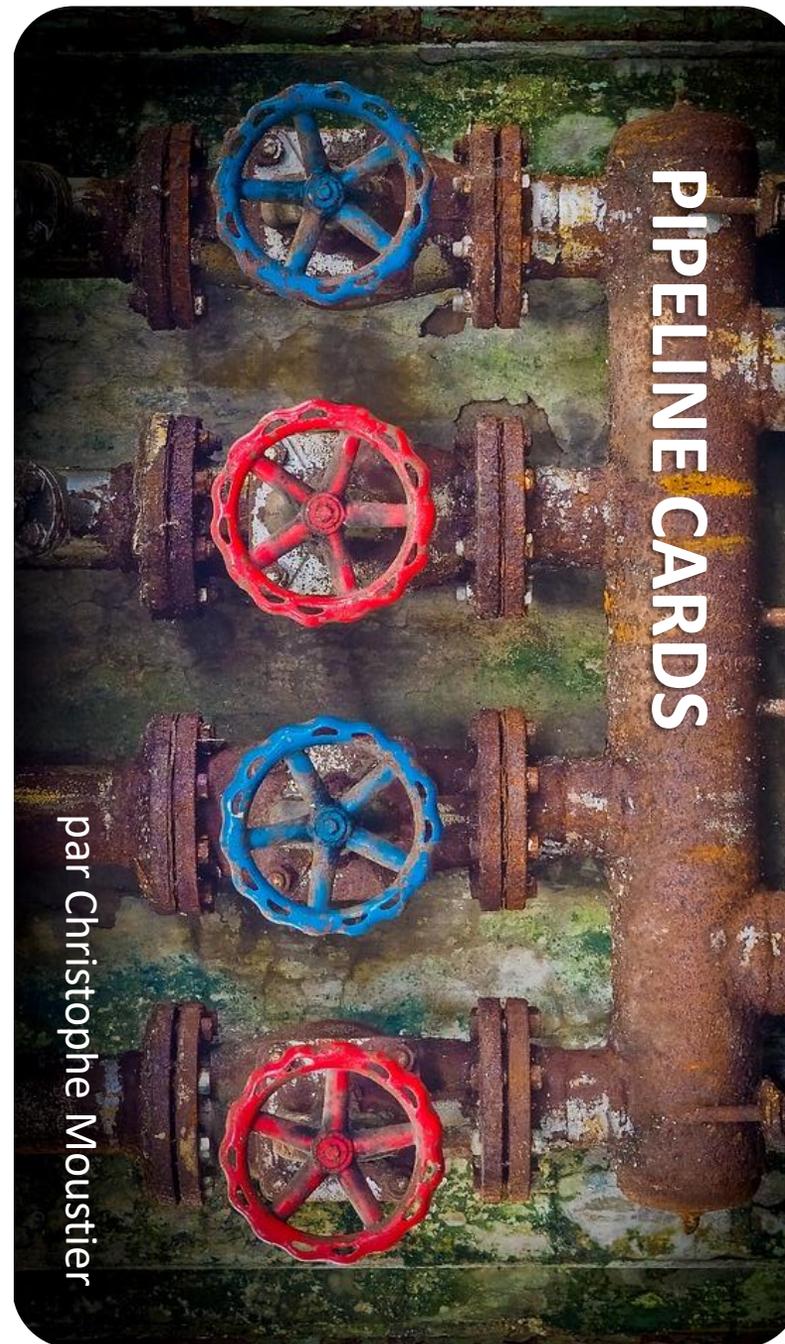
Permet de concilier des mises en production de faible qualité en fonction des circonstances avec les impératifs liés à l'exploitation

CI

CDv

CDp

CO





## TEST CONTINU

### Jidoka sur le pipeline

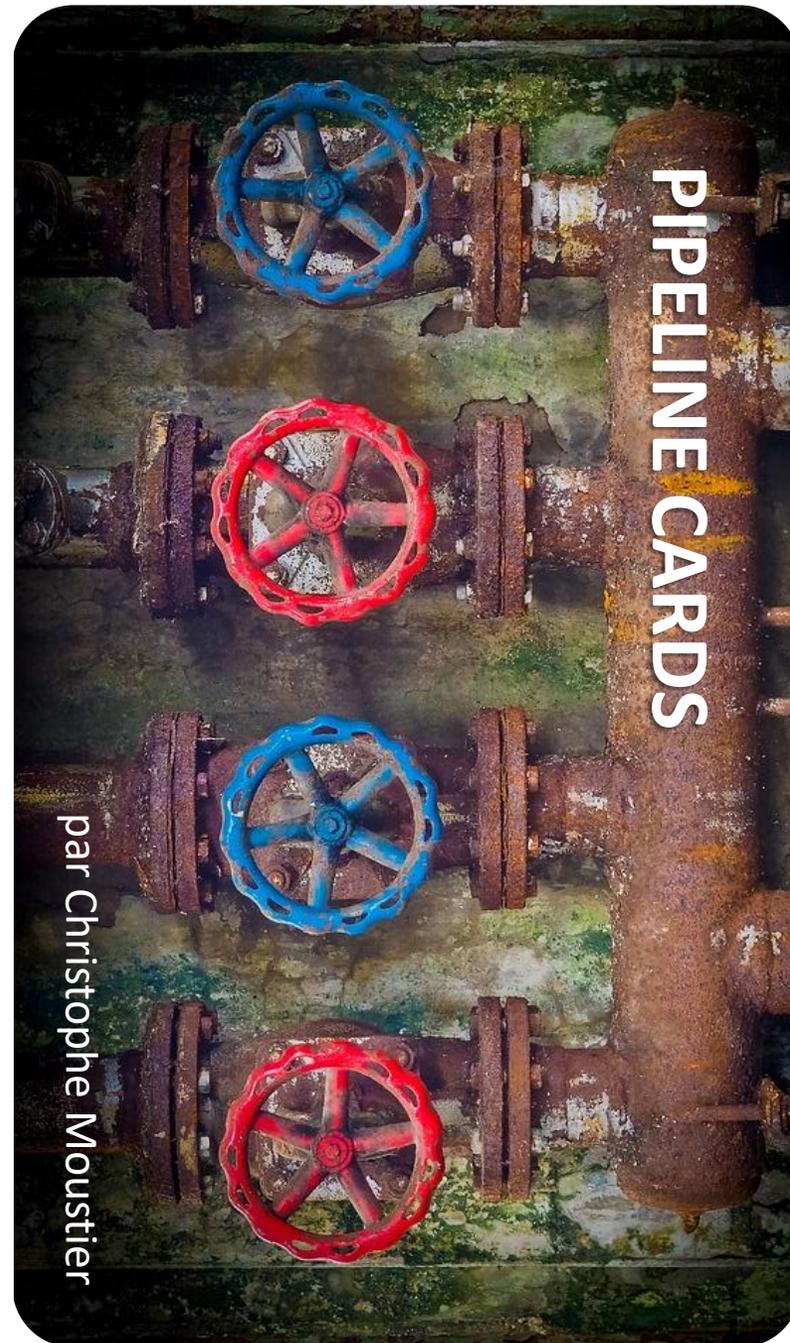
Introduire dans la conception du pipeline la tolérance aux pannes pour traiter automatiquement les problèmes et garantir le service avec correction des anomalies, quitte à mobiliser une partie de l'équipe (Andon)

CI

CDv

CDp

CO





## TEST CONTINU

### Jidoka sur les services rendus aux clients

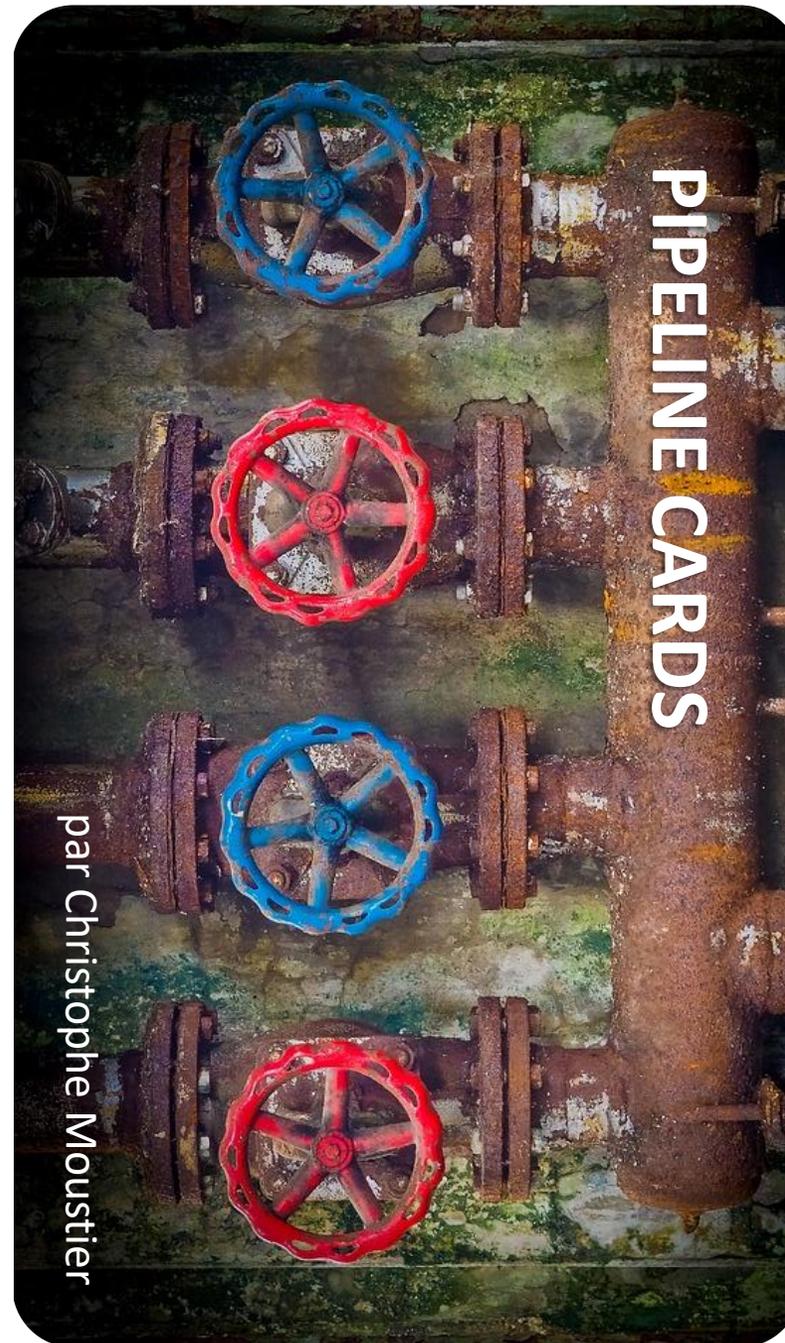
Introduire dans la conception de la solution la tolérance aux pannes pour traiter automatiquement les problèmes et garantir le service avec correction des anomalies, quitte à mobiliser une partie de l'équipe (Andon)

CI

CDv

CDp

CO





## TEST CONTINU

### Monitoring

Des seuils permettent de déclencher

- Des alertes pour actions humaines immédiates
- Des créations de tickets pour actions humaines à planifier
- Des logs pour faciliter le diagnostic ou la génération d'indicateurs

CI

CDv

CDp

CO





## TEST CONTINU

### Multiplier les types de tests sur la solution

Ne pas compter que sur le TDD, par exemple en utilisant les 4 quadrants du test agile

CI

CDv

CDp

CO





## TEST CONTINU

### Perturbation volontaire du système

Pour tester la résilience  
du système et pousser les  
mécanismes de tolérance  
aux pannes, le système  
peut déclencher des  
défaillances – ex. Chaos  
Monkey chez Netflix

CI

CDv

CDp

CO





## TEST CONTINU

### TDD

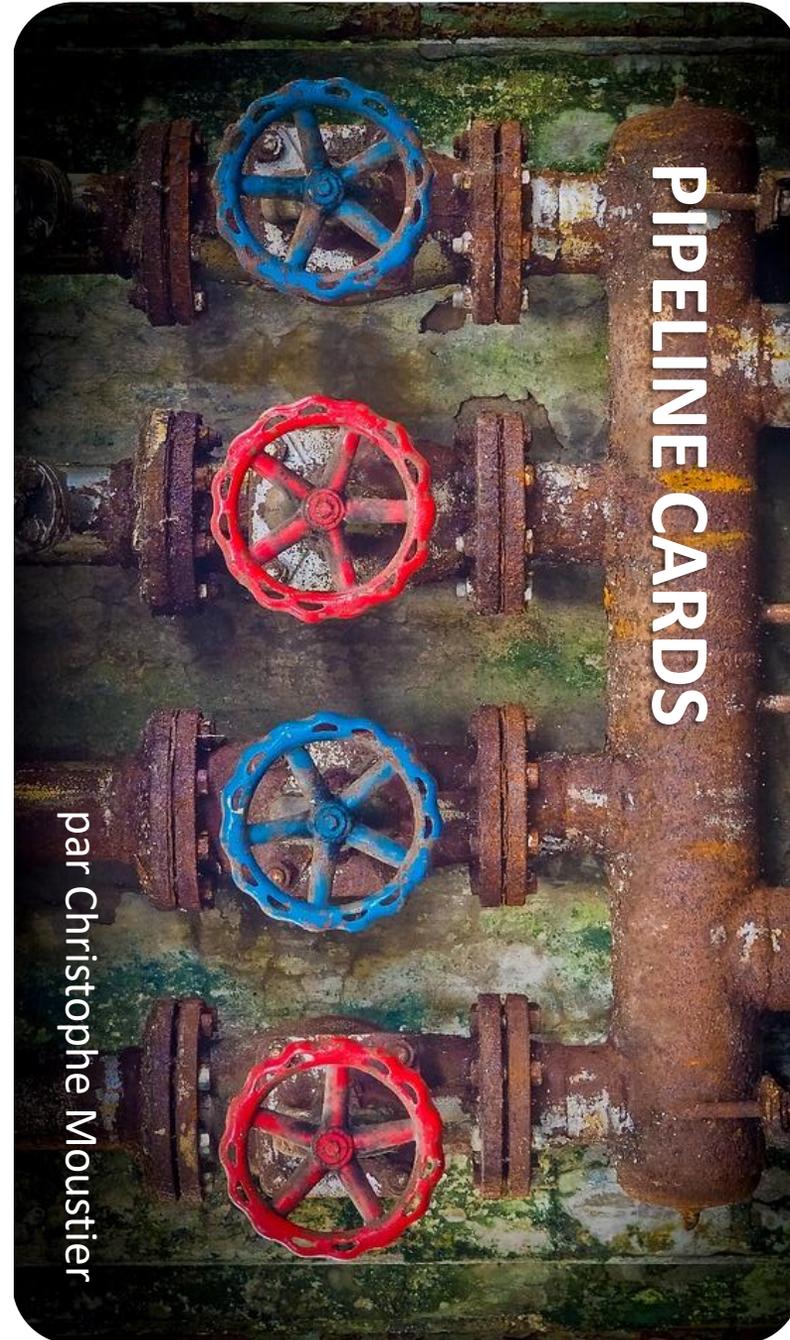
Cette pratique est prescrite par les frameworks d'agilité à l'échelle majeurs. Elle se base sur 3 états ROUGE / VERT / REUSINAGE

CI

CDv

CDp

CO





## TEST CONTINU

### Sélection automatique des tests dans le pipeline

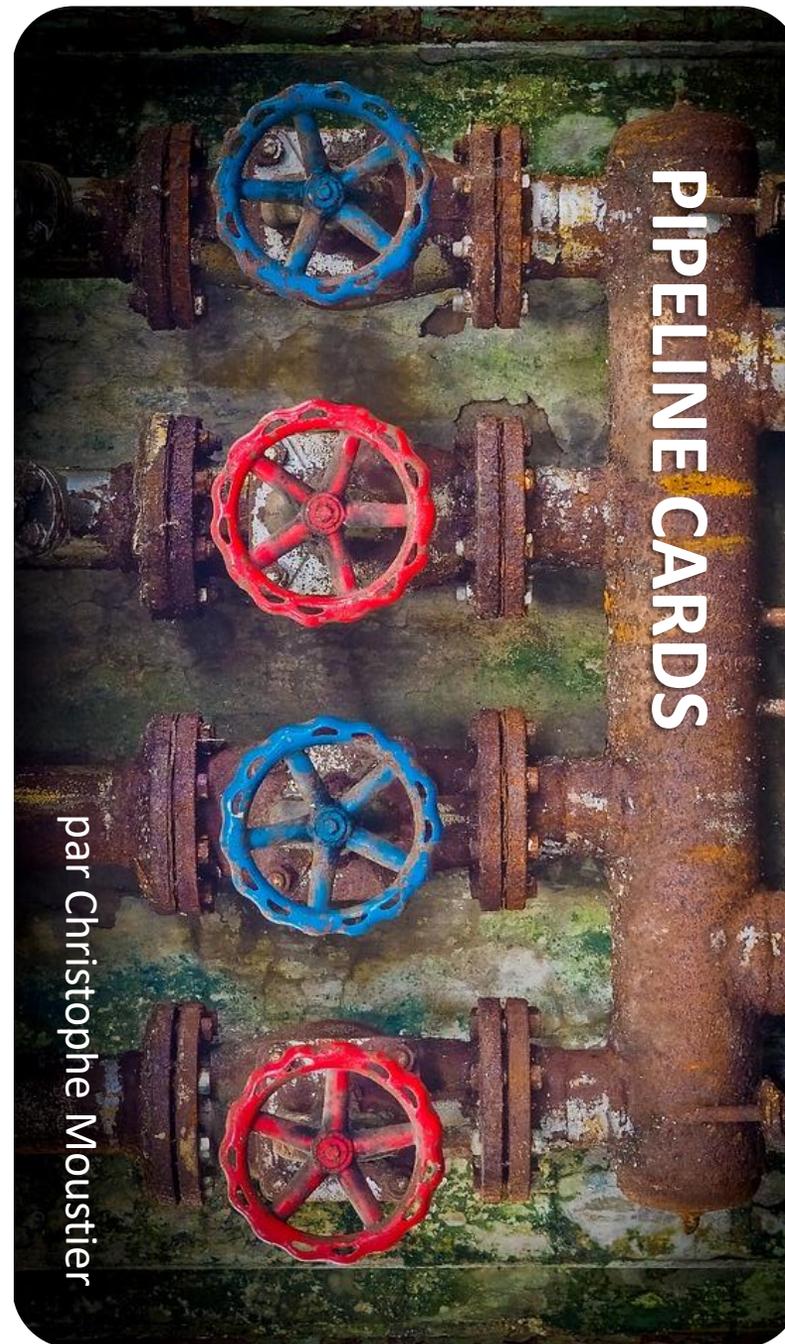
Pour permettre un temps  
de traitement court du  
pipeline, le RBT est  
appliqué aux tests  
automatisés disponibles

CI

CDv

CDp

CO





## TEST CONTINU

### ATDD

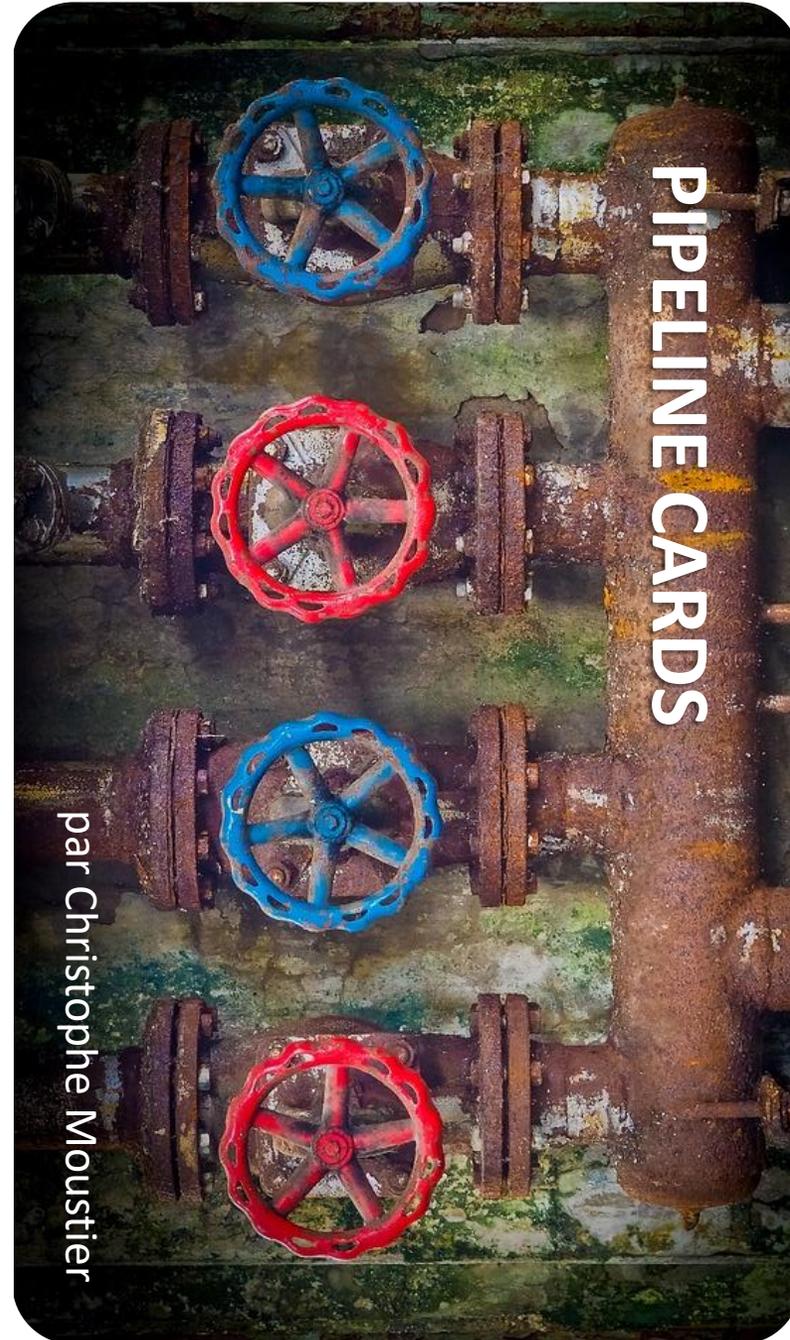
A partir des exemples qui permettent la validation des US, des tests automatisés sont créés pour générer le code de production (idéalement en mode TDD)

CI

CDv

CDp

CO





## TEST CONTINU

### Harnais de tests

Les composants de la solution sont dotés de bouchons et de scripts capables de les piloter.

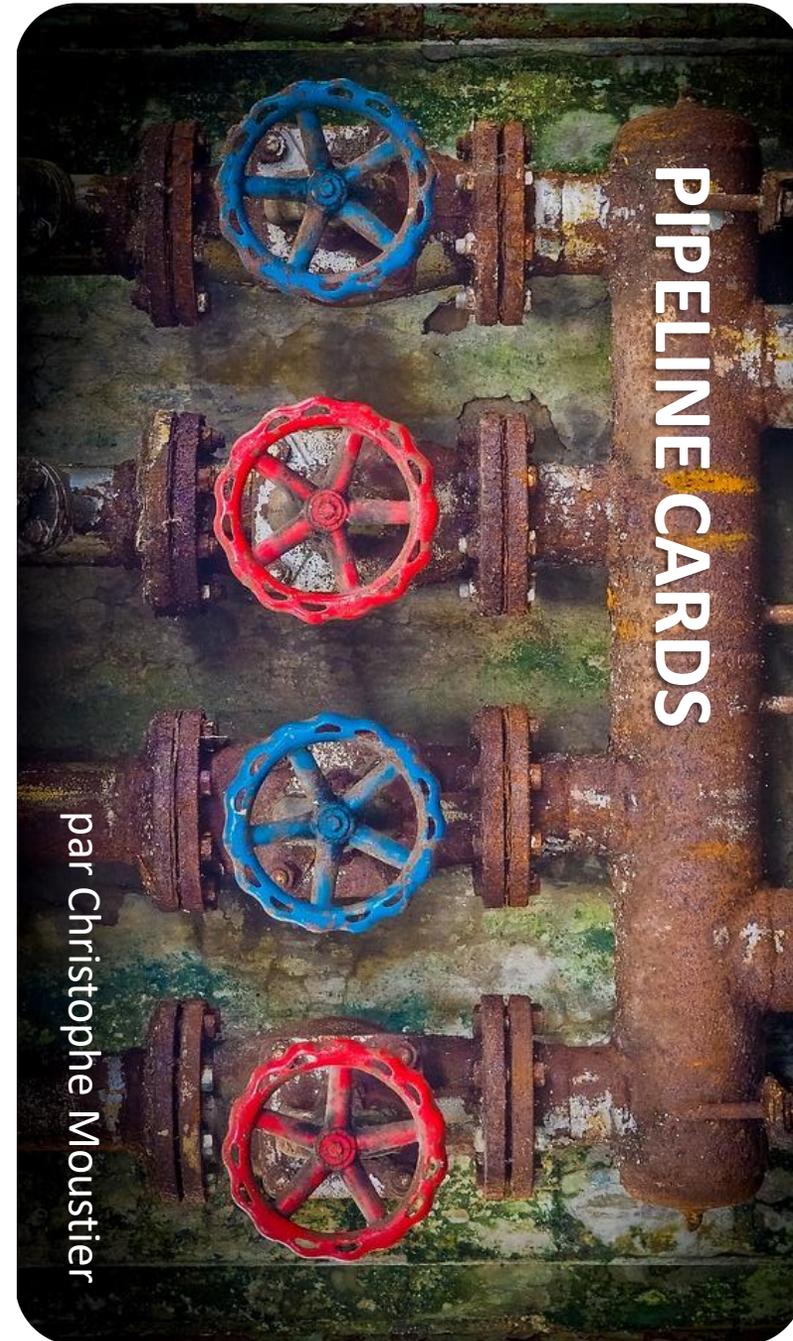
Ces harnais vont permettre des tests FIRST pour le pipeline

CI

CDv

CDp

CO





## TEST CONTINU

### Tests non fonctionnels

Certains tests non fonctionnels sont automatisables (ex. qualimétrie du code, failles de sécurité, tests d'intrusion, *etc.*) et peuvent être lancés par le pipeline

CI

CDv

CDp

CO





PANTESTING



## Cartographie des écocycles

Identifier les écocycles en jeu avec les connexions en cours de développement et imaginer avec les besoins de l'équipe les évolutions à apporter lors de l'itération suivante

CI

CDv

CDp

CO





PANTESTING



## Equipes X-Teams

Elles reposent sur

- Un réseautage interne
- Un réseautage externe
- Un niveaux extensibles (noyau, opérationnels, externe)
- Une appartenance flexible entre les niveaux
- Un mécanisme transparent

CI

CDv

CDp

CO





## PANTESTING

### Types d'ambidextries organisationnelles

L'organisation est partagée entre exploration et exploitation, l'ambidextrie.

On en trouve 4 formes :

- Harmonique : les 2 en même temps
- Cyclique : alternance de longues phases
- Partitionnée : harmonique localement
- Mutuelle : cyclique localement

CI

CDv

CDp

CO





## PANTESTING

### Priorisation WSJF des écocycles

Le développement des écocycles se priorise par un WSJF

$$WSJF = \frac{V+D+R}{C}$$

- V = Impact sur valeur business qualifiée en termes de mise à disposition ou d'absence
- D = impact du délai de mise en place
- R = risque ou opportunité pour le métier
- C = taille de l'écocycle

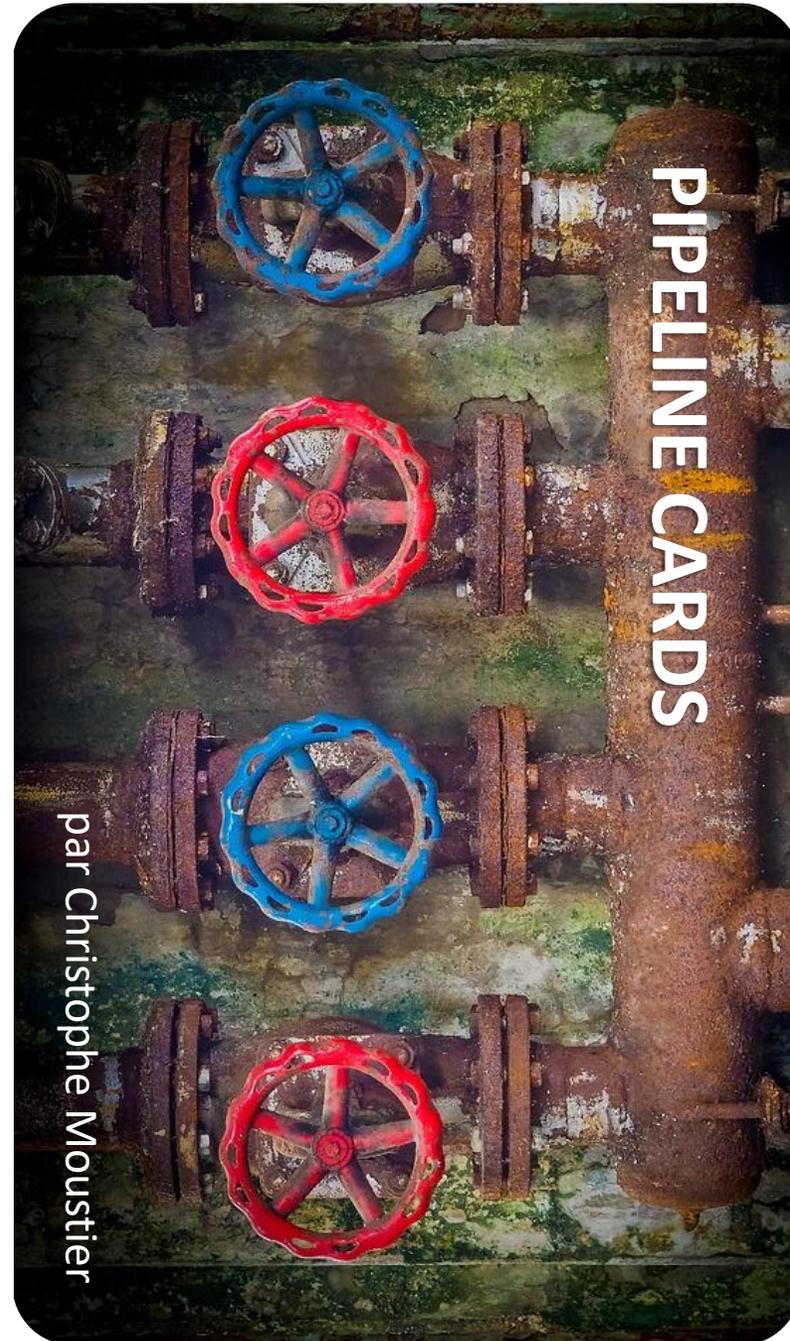
Valeurs [1 à 20] estimées en mode Poker Planning avec l'équipe

CI

CDv

CDp

CO





PANTESTING

## Amélioration de la testabilité

Les connexions vers d'autres écocycles permettent de trouver des contraintes supplémentaires et des opportunités de testabilité

CI

CDv

CDp

CO





## PANTESTING

### Mesure de connexion ASCCR inter-écocycle

La qualité des connexions  
vers les écocycles  
d'autres équipes est  
essentielle. ASCCR donne

#### 5 attributs

- Attitude (état d'esprit des partenaires)
- Structure (qualité des réunions, processus, etc.)
- Contenu
- Communication
- Relations

CI

CDv

CDp

CO





## PANTESTING

### Tableau +/- $\Delta$ pour gérer les doubles boucles d'apprentissage

Monitoring quotidien des doubles boucles d'apprentissage. Pour chaque double boucle, on note dans la colonne « + » ce qui marche, dans «  $\Delta$  » ce qui doit être amélioré

CI

CDv

CDp

CO





PANTESTING

## Historique de crédit pour gérer les doubles boucles d'apprentissage

Chaque itération, l'équipe  
note de 1 à 5 la facilité  
avec laquelle les choses  
se passent sur une  
double boucle pour  
approfondir ou diversifier

CI

CDv

CDp

CO





PANTESTING

## Atelier Panarchy des Liberating Structures

Atelier pour comprendre  
comment des systèmes  
intégrés interagissent,  
évoluent, diffusent  
l'innovation et se  
transforment

<https://www.liberatingstructures.fr/panarchy/>

CI

CDv

CDp

CO





PANTESTING

## Atelier Ecocycle des Liberating Structures

Atelier pour analyser  
l'intégralité des activités  
et des rapports d'un  
groupe pour identifier les  
obstacles au progrès et  
les opportunités  
d'évolution

<https://www.liberatingstructures.fr/ecocycle-planning/>

CI

CDv

CDp

CO





## PANTESTING

### Niveaux de connexion inter-écocycle

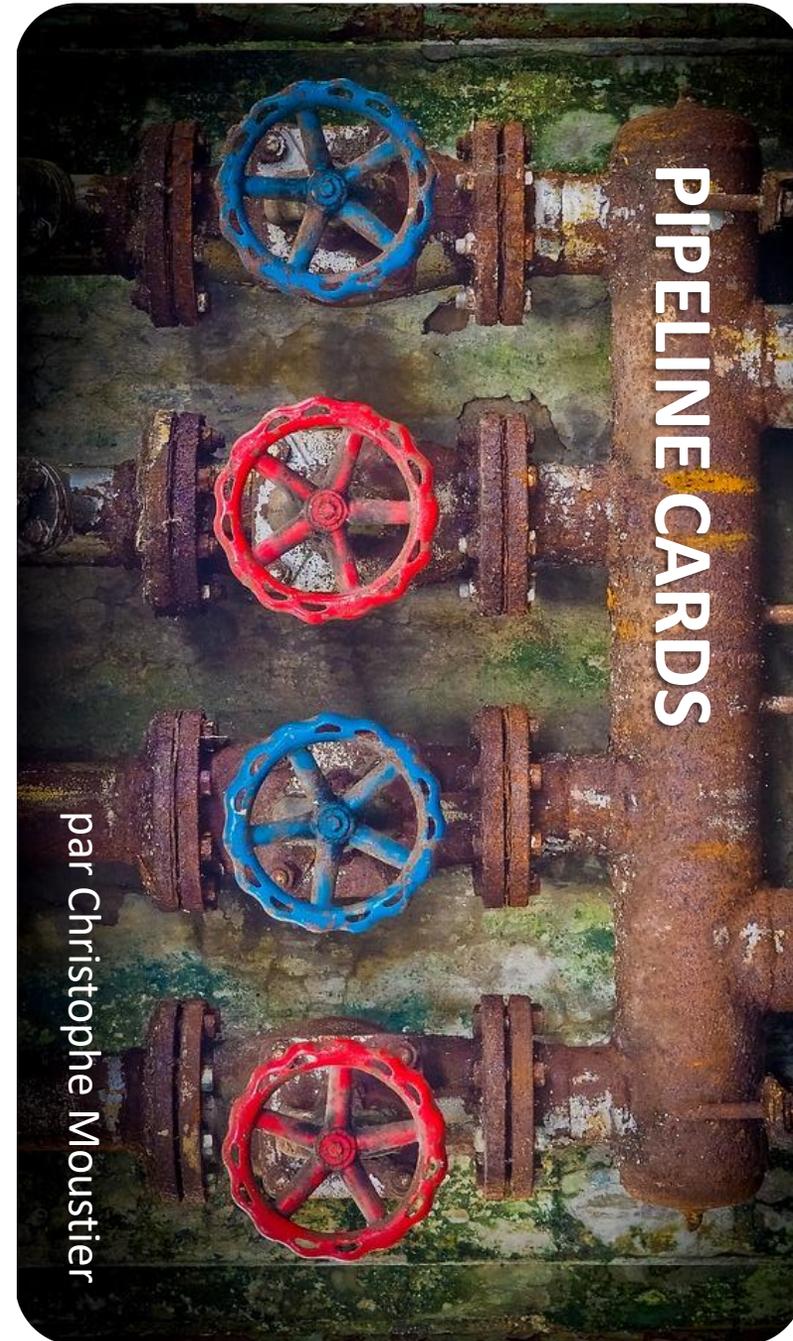
- 1. **Identifier** l'écocycle dont on dépend et amorcer la connexion
- 2. **Développer** les moyens de connexion entre les écocycles
- 3. **Organiser** et intégrer au moins les changements de phases  $\alpha$  et  $\Omega$  par la testabilité
- 4. **Anticiper** sur les phases de l'autre écocycle
- 5. **Fusionner** les écocycles

CI

CDv

CDp

CO





## PANTESTING

### Rôles du Pantesting

- **Collaborateurs** : Ambassadeurs, Yokoten, identifier des doubles boucles, plus de testabilité
- **Equipes** : Exploite / Anticipe les statuts des écocycles
- **Managers** : Améliore l'environnement
- **Communautés** : Hub de connexions

CI

CDv

CDp

CO





## PANTESTING

### Suivi du Pantesting

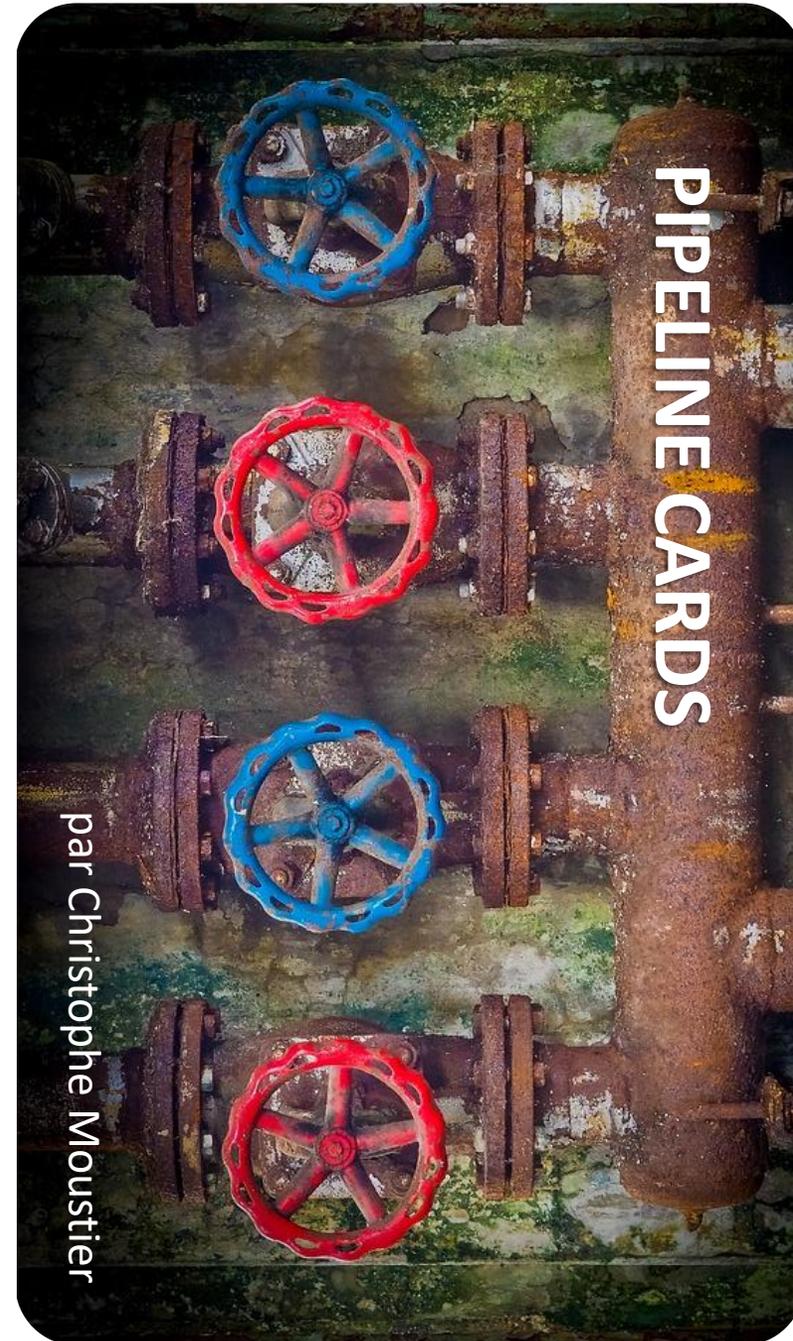
Indicateurs & objectifs  
basés sur les OKR  
Les objectifs doivent  
incorporer des facteurs  
humains et de qualité des  
processus (ex. BSC,  
EFQM)

CI

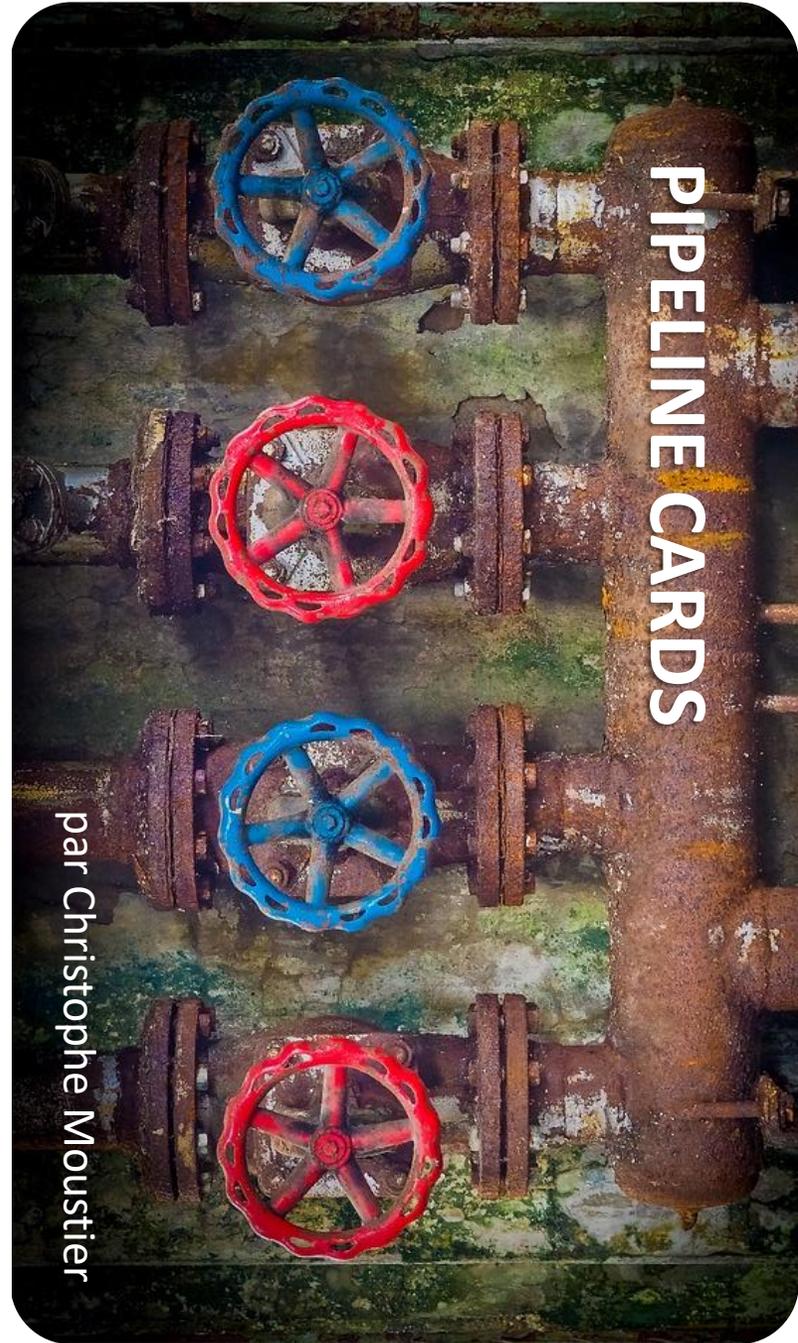
CDv

CDp

CO



# PIPELINE POWER CARD



par Christophe Moustier

# Print



Copies: 1

## Printer

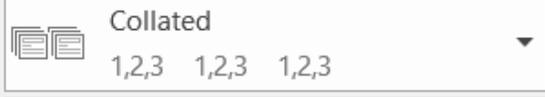
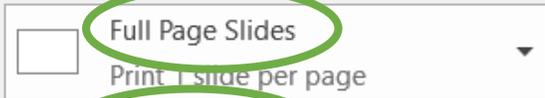


Printer Properties

## Settings



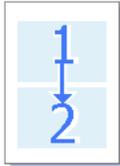
Slides:



[Edit Header & Footer](#)

Frequently Used Settings Detailed Settings Configuration/About

Current Setting: User Setting



One Click Preset List: Factory Default

Job Type: Normal Print

Document Size: A4 (210 x 297 mm)

Print On: Same as Original Size

Paper Type: Plain & Recycled

Orientation:  Portrait  Landscape

Layout: 2 Pages per Sheet

2 sided: Off

Page Order: Left-Right/Top-Bottom

Booklet: Off

Staple: Off

Punch: Off

Color/ Black and White: Color

Copies: (1 to 999) 1

OK Cancel Help

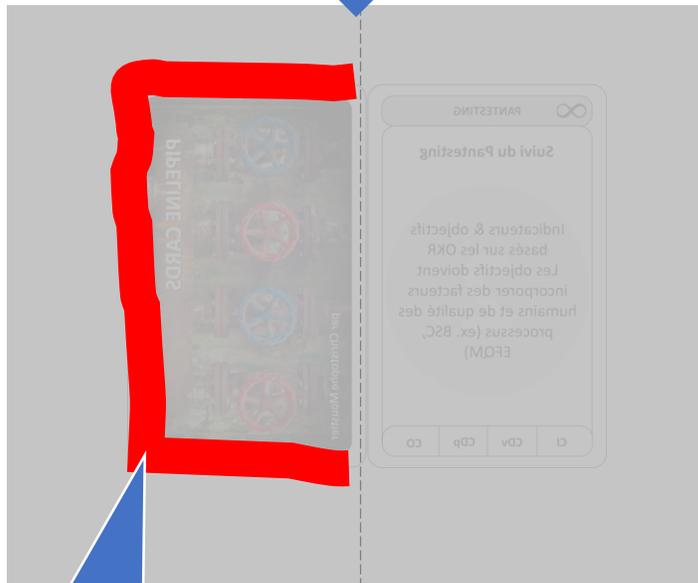
*Note: l'impression devrait donner 2 cartes par feuille A4*

1



Plier ici

2



Mettre de la colle ici

3

Couper ici

