PRINCIPES DEVOPS

Dawan
Training is our way

- Voie 1: le Flux
  - Le Flux: Pratiques
- Voie 2: La Rétroaction
  - La Rétroaction: Pratiques
- Voie 3: Formation & Experimentation Continues
  - La Formation & Expérimentation continue: Pratiques

Introduction: les 3 voies du DevOps

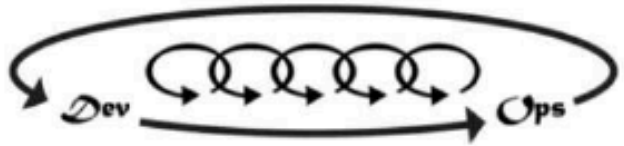
- Concept émis par **Gene kim**, CTO de Tripwire et chercheur, en *2013*, dans son livre **« the phoenix project »**

> “ Les 3 voies sont le schéma directeur de la *mise en œuvre d’une démarche DevOps* ”

Ces 3 voies sont

```
- le Flux
- la Rétroaction
- la Formation & Experimentation
```

Les Three ways



Première voie	Deuxième voie	Troisième voie
Flux	Feedback	Experimentation et apprentissage continus
Comprendre et améliorer le flux de travail (de gauche à droite)	Créer des boucles de feedback permettant une amélioration continue (de droite à gauche)	Créer une culture qui favorise : • L'expérimentation, la prise de risques et tirer des leçons des échecs • Comprendre que la répétition et la pratique est la condition préalable à la maîtrise

Voie 1: le Flux

La première voie : le flux

- Comprendre le flux de travail
- Améliorer le flux en identifiant puis en supprimant les contraintes
- Ne jamais transmettre un défaut connu vers l'aval
- Ne jamais laisser une optimisation locale entrainer la dégradation globale
- Atteindre une véritable compréhension de l'ensemble du système

La première voie vise à fluidifier le travail afin qu'il s'écoule rapidement de gauche à droite.

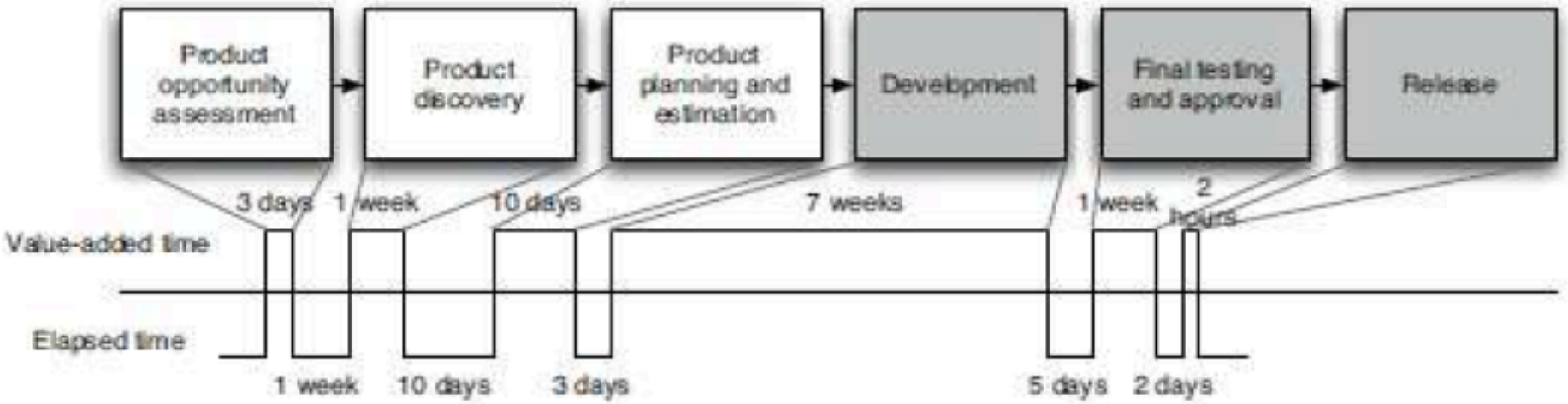
Le Flux: Pratiques

Value Stream Mapping (VSM)

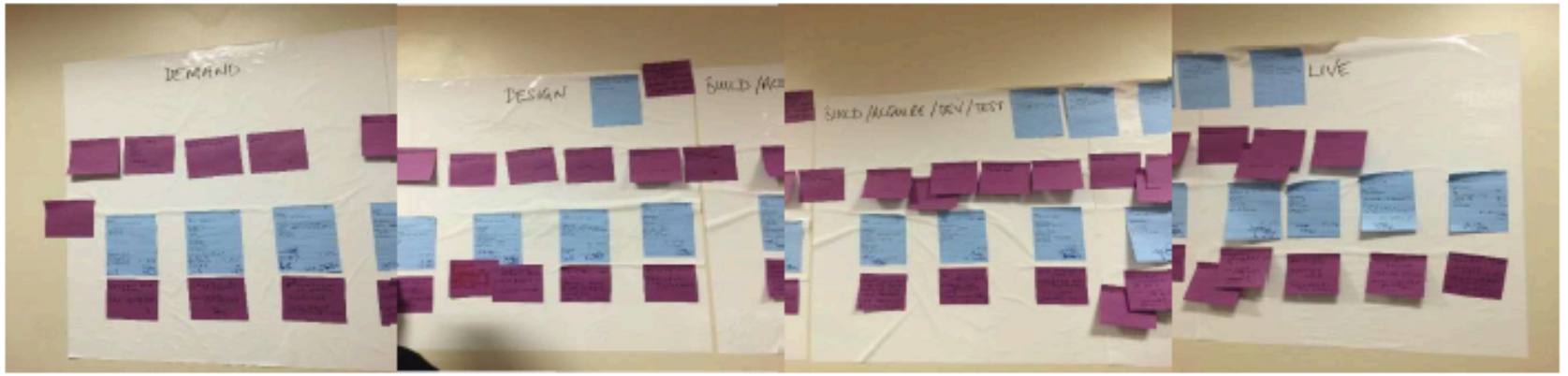
Value stream mapping est un outil Lean qui décrit le flux d'informations, de matériaux et de travail entre silos fonctionnels, en mettant l'accent sur la quantification et l'élimination du gaspillage, y compris en termes de temps et de qualité.

- Une chaîne de valeur est la séquence d'activités requises pour concevoir, produire et livrer un produit ou un service spécifique
- Les chaînes de valeur couvrent généralement plusieurs processus
- La cartographie des chaînes de valeur permet aux équipes fonctionnelles transverses de :
  - Voir tout une chaîne de valeur du point de vue du travail et des flux d'informations
  - Identifier les zones de gaspillage sans valeur qui pourraient être éliminées dans le but d'améliorer la chaîne et de générer une plus grande valeur
  - Identifier, hiérarchiser et mesurer les améliorations

Exemple de Value Stream Maps



Source :
Jez Humble - *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*



Source : Ranger4

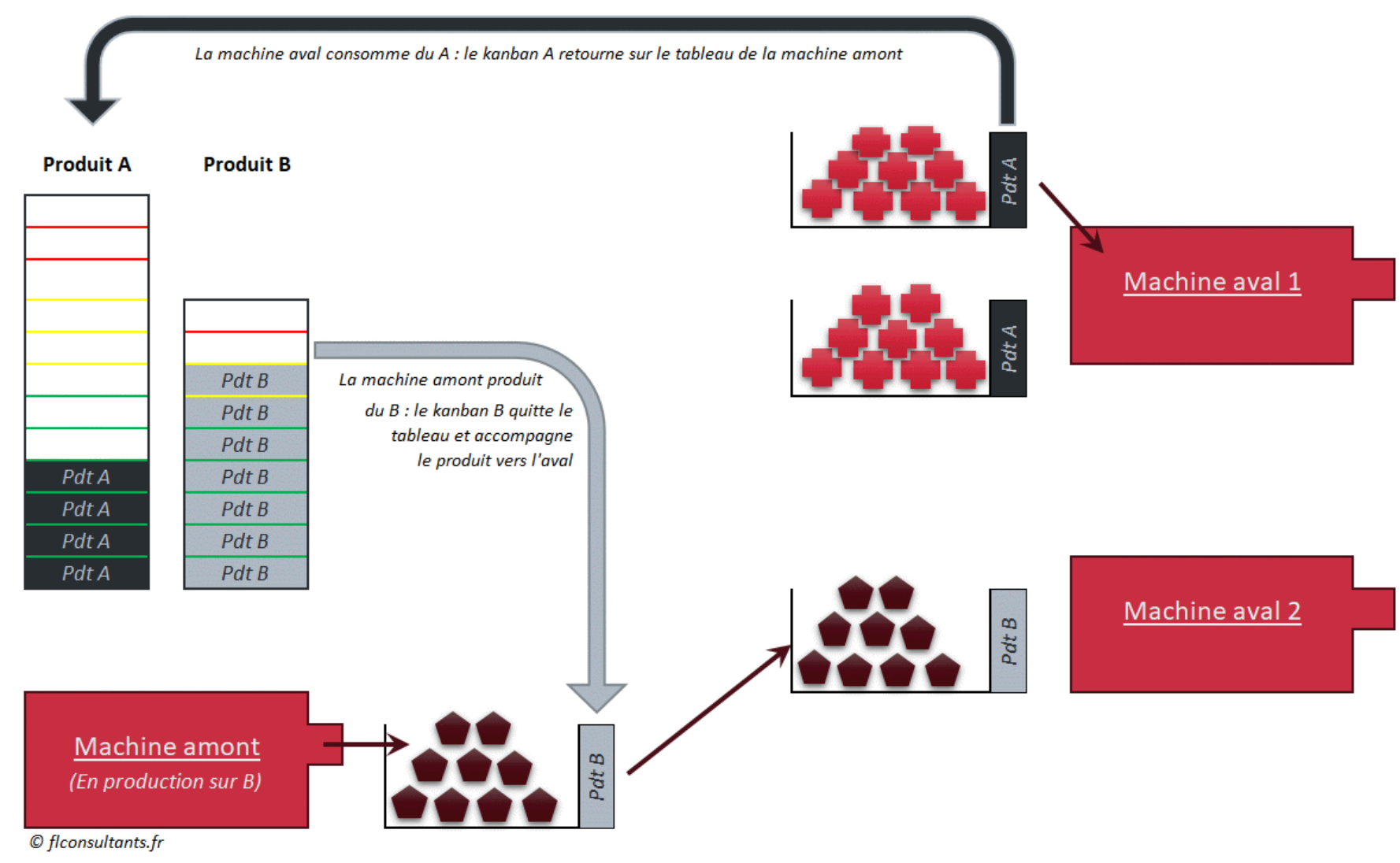
Kanban

Kanban est une approche de visualisation du travail qui permet de mieux « tirer » le flux de travail au sein d'un processus à un rythme raisonnable.

- Visualise et gère le flux de travail
- Tire le travail pour les équipes quand elles sont prêtes
- Permet aux gens de travailler en collaboration pour améliorer le flux
- Mesure la vélocité de l'équipe (quantité de travail effectué lors d'une itération)
- Réduit les temps morts et les pertes dans un processus

- Rend le travail visible
- Rend les politiques explicites
- Limite les travaux en cours (WIP) à la capacité

KANBAN historique (LEAN comme VSM)



Théorie des contraintes

Une approche permettant d'identifier le plus important facteur limitant (c.à-d. la contrainte) qui retarde l'atteinte d'un objectif et de systématiquement optimiser cette contrainte jusqu'à ce qu'elle ne soit plus le facteur limitant.

La théorie des contraintes reconnaît que :

- Chaque processus a au moins une contrainte ou un goulot d'étranglement qui affecte sa capacité à atteindre son objectif.
- Le processus pourra au maximum atteindre la capacité de ses contraintes et pourra être efficace à la limite de son maillon le plus faible.
- L'optimisation des contraintes est le moyen le plus rapide et le plus efficace d'améliorer l'ensemble du processus ou du système

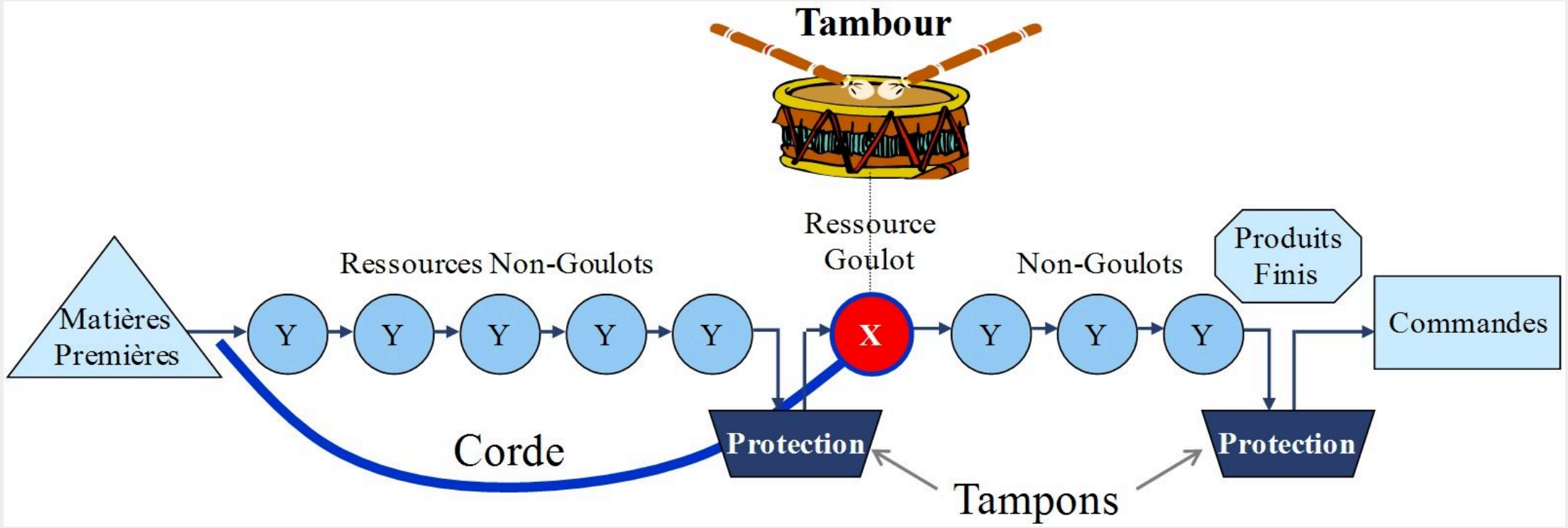
La théorie des contraintes a été introduite dans le livre 'The Goal' par Eliyahu M. Goldratt.

Contraintes courantes

- Délais de développement
- Création d'environnement (test, simulation, production, etc.)
- Déploiement de code
- Mise en place et exécution des tests
- Évaluations de sécurité ou de QA
- Dépendance à l'architecture
- Gestion des produits
- Processus complexes ou bureaucratiques

Théorie des contraintes: Historique

- méthode: **DRB**: Drum Rope Buffer



Integration continue

L'intégration continue (CI) est une pratique de développement qui oblige les développeurs à « committer » du code dans un référentiel partagé (maître/tronc) au moins une fois par jour.

- Chaque archivage est validé par
  - Une version automatisée
  - Des tests unitaires, d'intégration et de réception automatisés
- Dépend de normes de codage uniformes
- Nécessite des référentiels de contrôle de version et des serveurs CI pour collecter, construire et tester le code « committé »
- Fonctionne dans des environnements semblables à la production
- Permet une détection précoce et une correction rapide des erreurs dues aux modifications de code avant de passer en production

Bien qu'elles soient principalement associées au développement logiciel agile, les approches en cascade peuvent également tirer parti de l'intégration continue et des pratiques du test-driven development (TDD).

Intégration Continue: autre définition

- *processus liant pratiques et solutions* concourant à
  - éliminer les erreurs - **« régressions »** OU
  - amener un code à l'état de **livrable**, i.e *démontrable à un client*
- exécuté **périodiquement** sur un env. commun => branche GIT
- exemples:
  - tests
  - qualité
  - sécurité

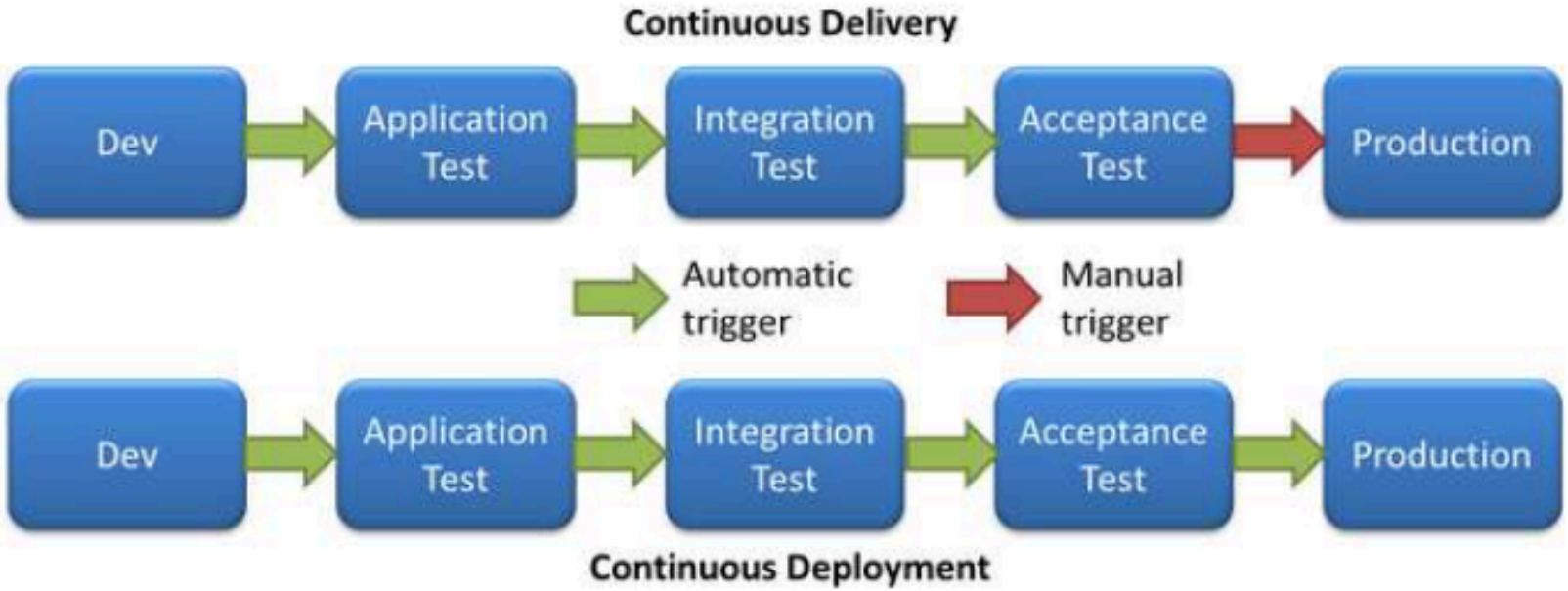
Livraison continue

Le livraison continue (CD) est une méthodologie qui vise à s'assurer que le logiciel est ***toujours dans un état livrable*** tout au long de son cycle de vie.

- Fait passer l'intégration continue au niveau supérieur
- Fournit un feedback rapide et automatisé sur l'état de préparation à la production d'un système
- Donne la priorité au maintien du logiciel livrable / déployable par rapport au travail sur les nouvelles fonctionnalités
- S'appuie sur un pipeline de déploiement permettant des déploiements à la demande en un clic
- Réduit les coûts, les délais et les risques liés aux modifications incrémentielles

Extrait du rapport DevOps 2017

Livraison continue et déploiement continu



L'intégration continue est la pratique permettant la livraison continue de la valeur entre les mains des utilisateurs

Extrait de Mirco Hering : notafactoryanymore.com, auteur de 'DevOps for the Modern Enterprise'

ChatOps

Group chat client + chat bots = développement, livraison et support axés sur la conversation

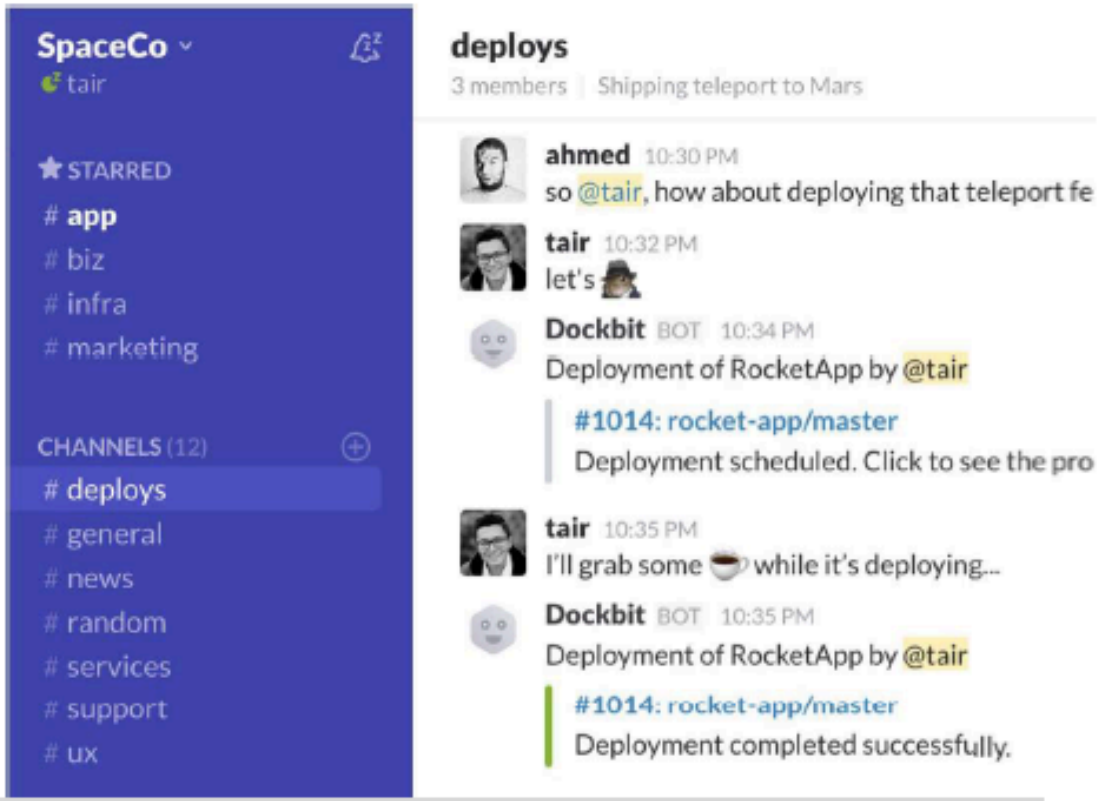
Stride

Microsoft Teams

slack

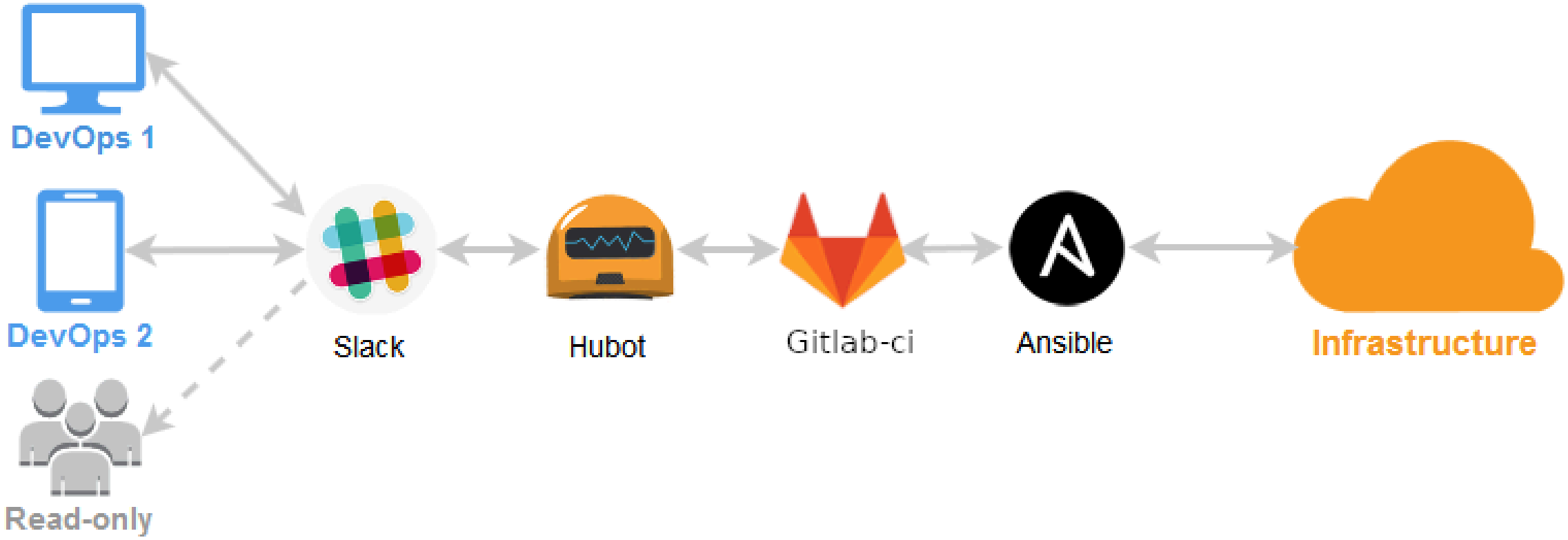
Mattermost

Google Hangouts



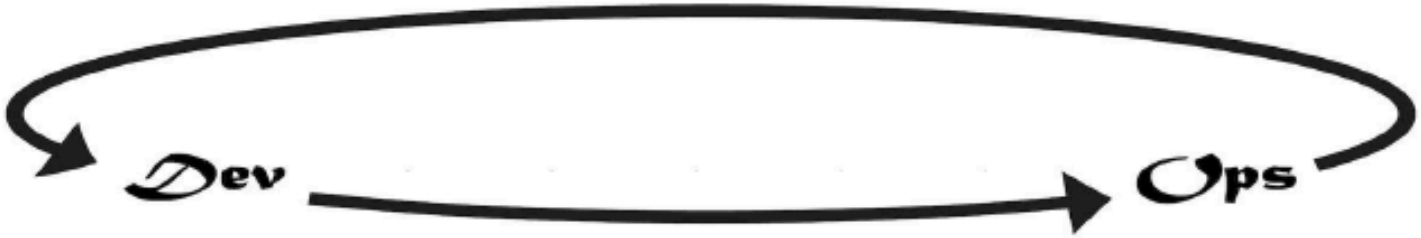
La transparence de ChatOps raccourcit les boucles de feedback, améliore le partage d'informations, améliore la collaboration en équipe et permet une formation transversale. Il peut également être utilisé pour diminuer le MTTR.

ChatOps: schéma classique



Voie 2: La Rétroaction

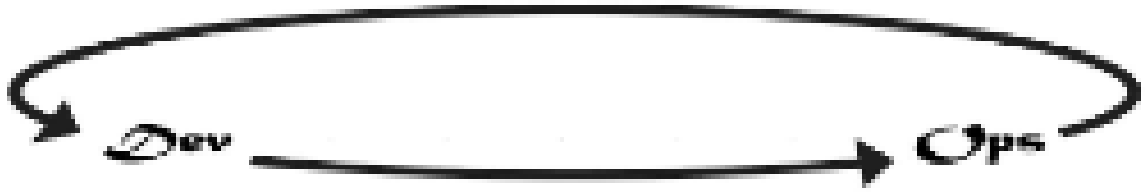
Deuxième voie : Feedback



- Comprendre et répondre aux besoins de tous les clients - internes comme externes
- Raccourcir et amplifier toutes les boucles de feedback
- Créer et intégrer des connaissances là où elles sont nécessaires

La deuxième voie vise à raccourcir et amplifier les boucles de feedback de droite à gauche afin que les corrections nécessaires puissent être apportées en permanence.

Exemples de boucles de feedback



- Tests automatisés (rapports)
- Revue par les pairs des changements en production
- Monitoring + observabilité
- Tableaux de bord
- Logs de production
- Indicateurs de processus
- Post-mortem
- Partage d'informations entre les pilotes d'exploitation
- Données de changements, incidents, problèmes et gestion connaissances

La Rétroaction: Pratiques

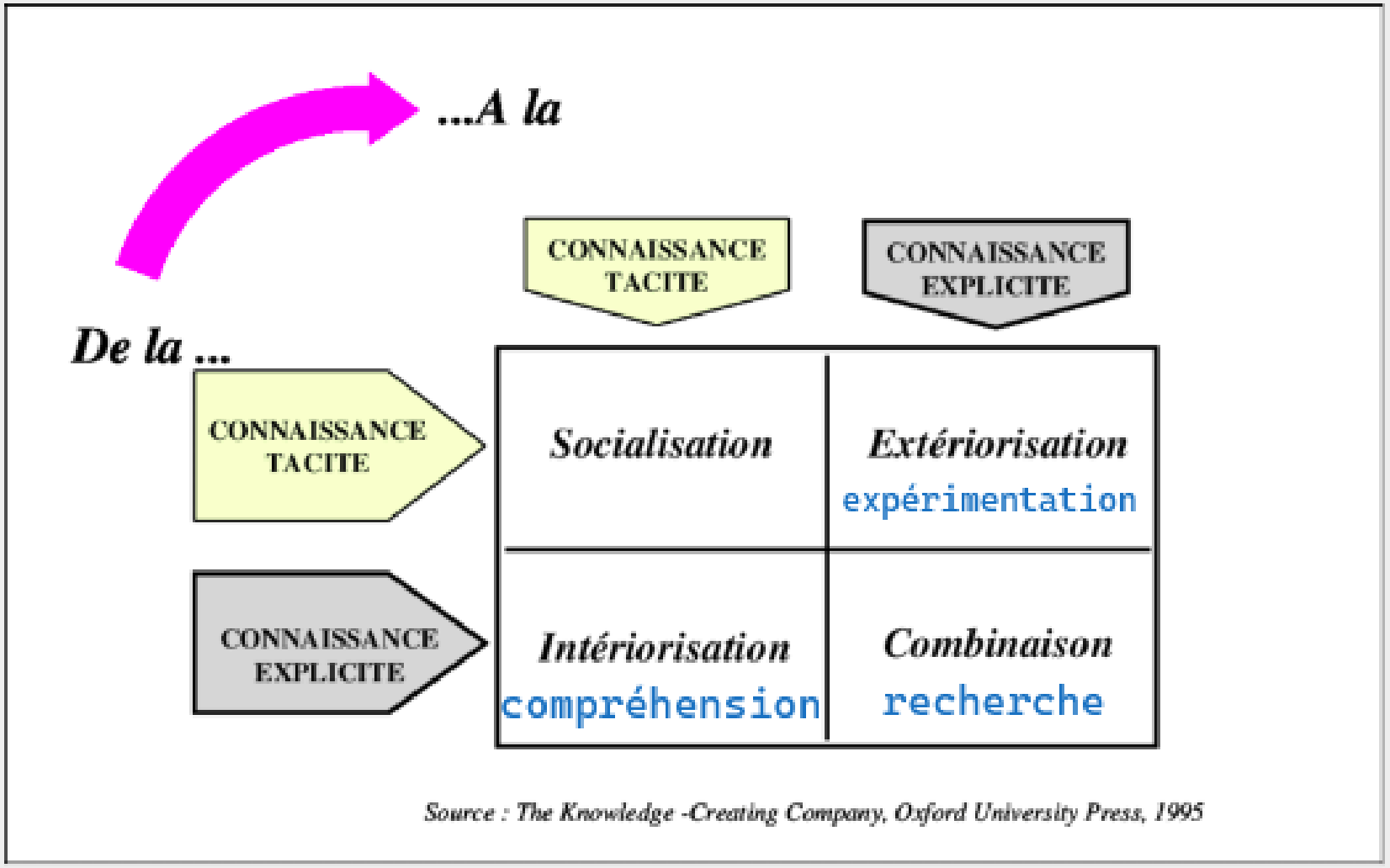
Gestion des connaissances

“ Issue de l’anglais *« knowledge management »* ou **KM** ”

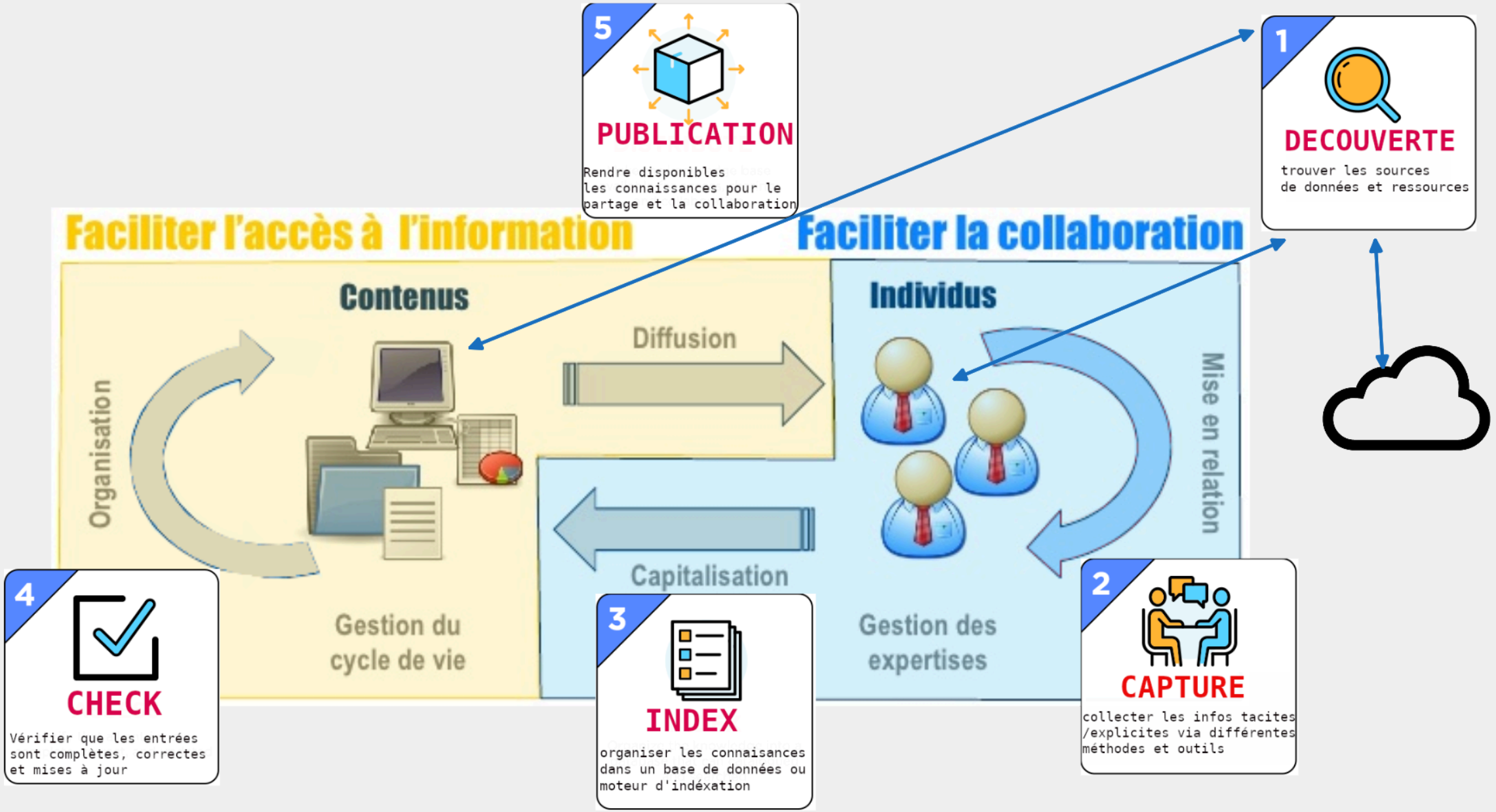
Vocable

Source d’information : origine humaine ou non d’une donnée
Donnée : *mesure* d’une réalité en fonction d’un étalon de référence
Information: *donnée* porteuse de sens, interprétation d’une donnée
Connaissance: utilisation à long terme d’une *information*
Raisonnement: traitement d’un résau d'*infos* et de *connaissances*
Compétence: corpus de *connaissances* pour agir ou décider
Savoir – faire : mise à profit des *compétences d’un métier*

partages de connaissances



Source : The Knowledge -Creating Company, Oxford University Press, 1995


5
PUBLICATION
Rendre disponibles les connaissances pour le partage et la collaboration
1
DECOUVERTE
trouver les sources de données et ressources
Faciliter l'accès à l'information
Faciliter la collaboration
Contenus
Individus
Diffusion
Organisation
Mise en relation
Capitalisation
Gestion du cycle de vie
Gestion des expertises
4
CHECK
Vérifier que les entrées sont complètes, correctes et mises à jour
3
INDEX
organiser les connaissances dans un base de données ou moteur d'indéxation
2
CAPTURE
collecter les infos tacites /explicites via différentes méthodes et outils

Exemple de Post-Mortem: le Retour d'expérience

> **REX** ou **RETEX**, une démarche qui permet d'*apprendre d'une expérience* afin de mieux *anticiper l'avenir*.

Objectifs:

Partager une vision globale commune de l'expérience.

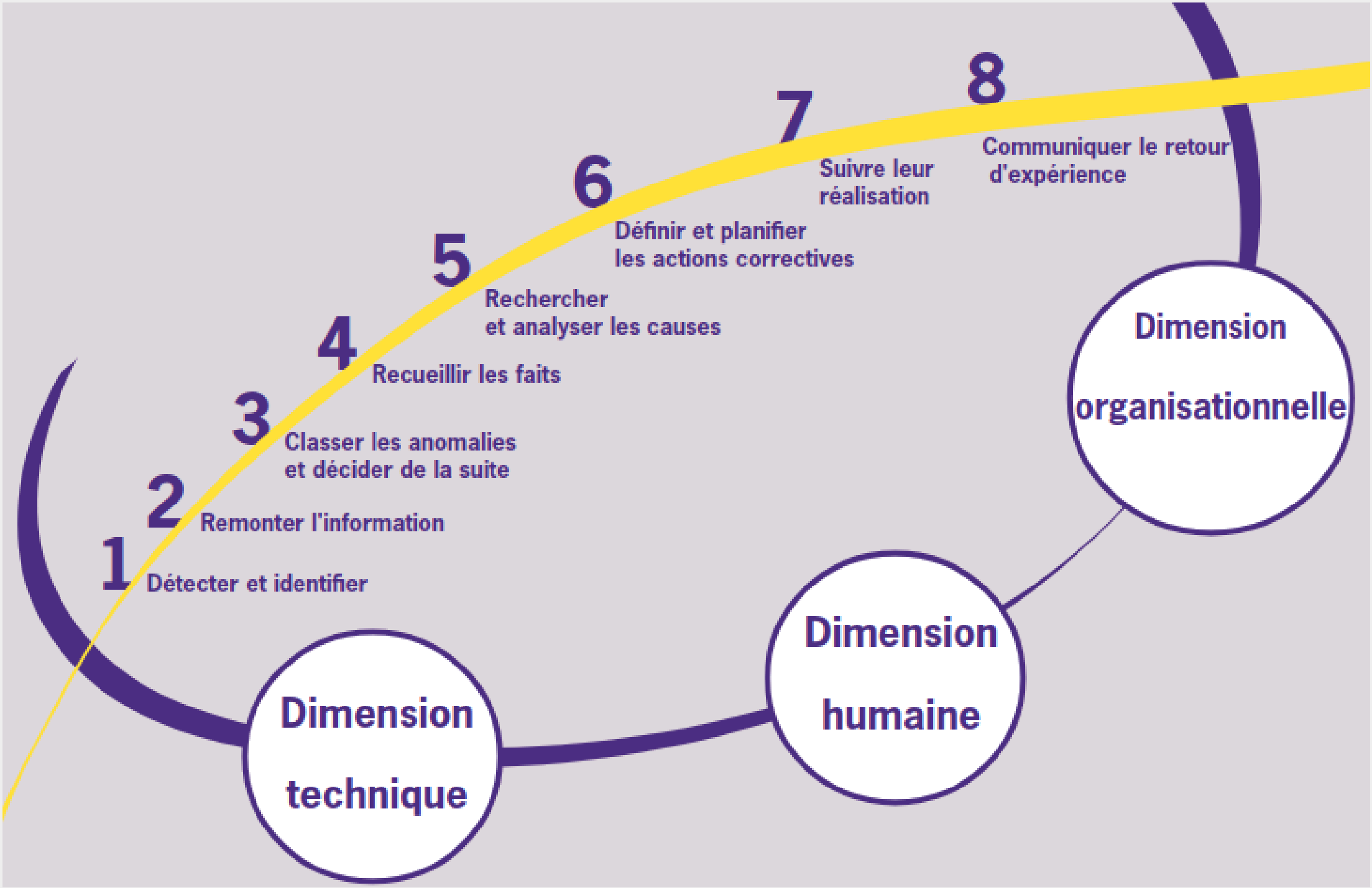
Renforcer les liens entre les acteurs

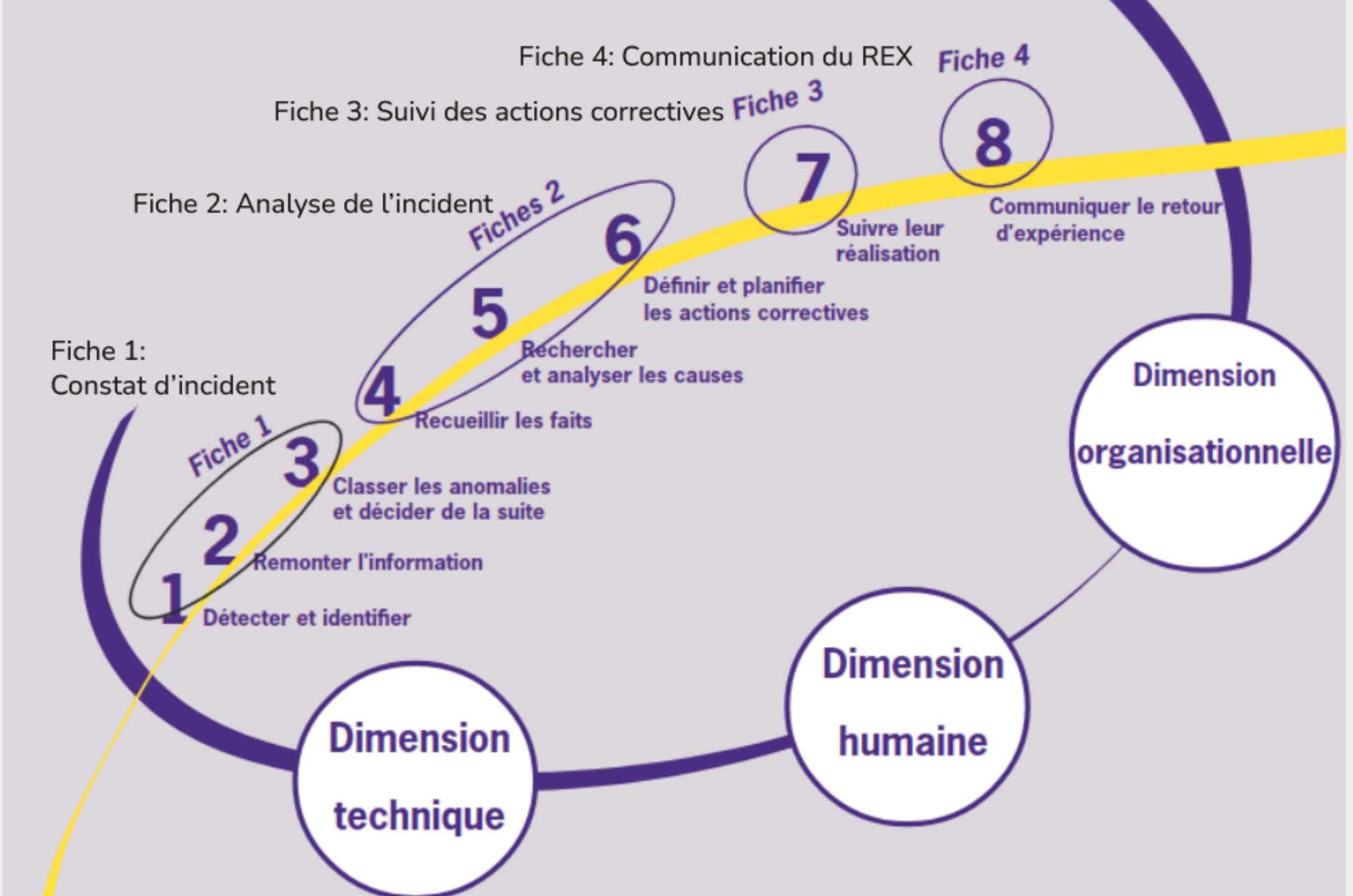
capitaliser les points positifs

Identifier les points négatifs et proposer des axes d'amélioration

Reconnaître le travail de chacun

valoriser l'expérience acquise pour la gestion des incidents futurs et ainsi améliorer les performances de l'entreprise.


1
Détecter et identifier
2
Remonter l'information
3
Classer les anomalies
et décider de la suite
4
Recueillir les faits
5
Rechercher
et analyser les causes
6
Définir et planifier
les actions correctives
7
Suivre leur
réalisation
8
Communiquer le retour
d'expérience
Dimension
technique
Dimension
humaine
Dimension
organisationnelle


Fiche 4: Communication du REX
Fiche 4
Fiche 3: Suivi des actions correctives
Fiche 3
Fiche 2: Analyse de l'incident
Fiches 2
Fiche 1:
Constat d'incident
Fiche 1
1
Détecter et identifier
2
Remonter l'information
3
Classer les anomalies
et décider de la suite
4
Recueillir les faits
5
Rechercher
et analyser les causes
6
Définir et planifier
les actions correctives
7
Suivre leur
réalisation
8
Communiquer le retour
d'expérience
Dimension
organisationnelle
Dimension
humaine
Dimension
technique

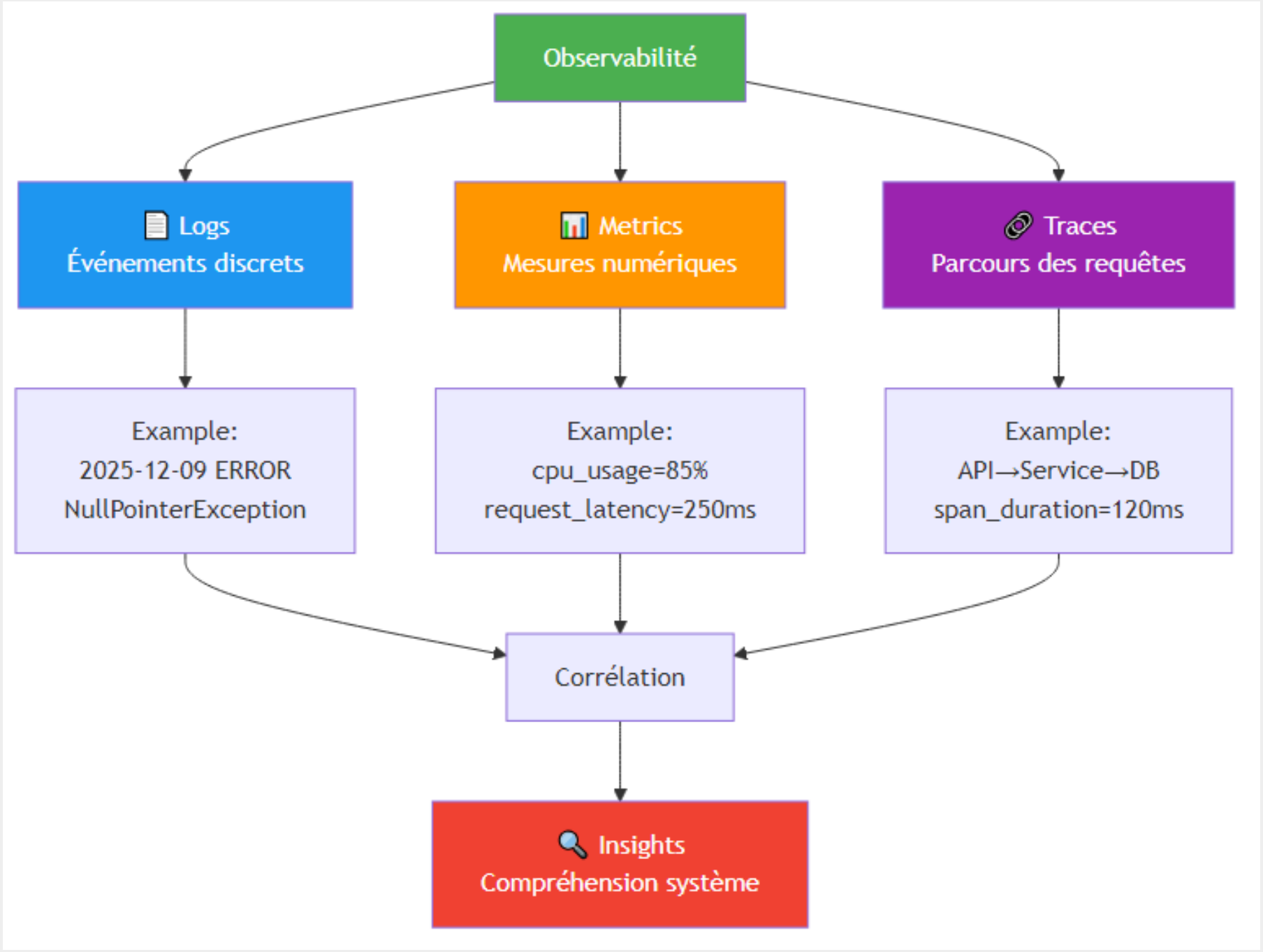
Observabilité

- **monitoring**:

“ surveillance continue et automatisée d'un système informatique pour *vérifier qu'il fonctionne correctement* et détecter les anomalies **==> voyants du tableau de bord (t°, vitesse)** ”

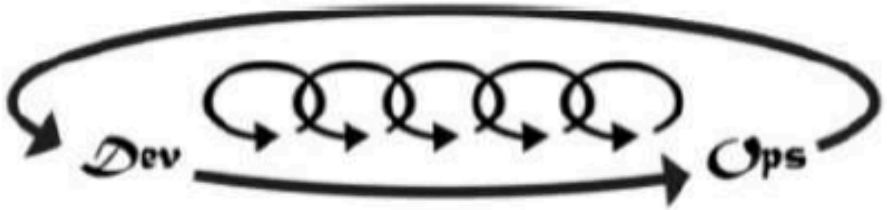
- **Observabilité**:

“ *capacité à comprendre l'état interne* d'un système en examinant ses sorties externes (logs, métriques, traces). C'est une propriété du système qui permet de diagnostiquer des problèmes sans avoir à le modifier **==> Boîte noire + tous les capteurs** ”


Observabilité
Logs
Événements discrets
Metrics
Mesures numériques
Traces
Parcours des requêtes
Example:
2025-12-09 ERROR
NullPointerException
Example:
cpu_usage=85%
request_latency=250ms
Example:
API→Service→DB
span_duration=120ms
Corrélation
Insights
Compréhension système

Voie 3: Formation & Experimentation Continues

La troisième voie : Experimentation et apprentissage continus



La troisième voie encourage une culture qui favorise :

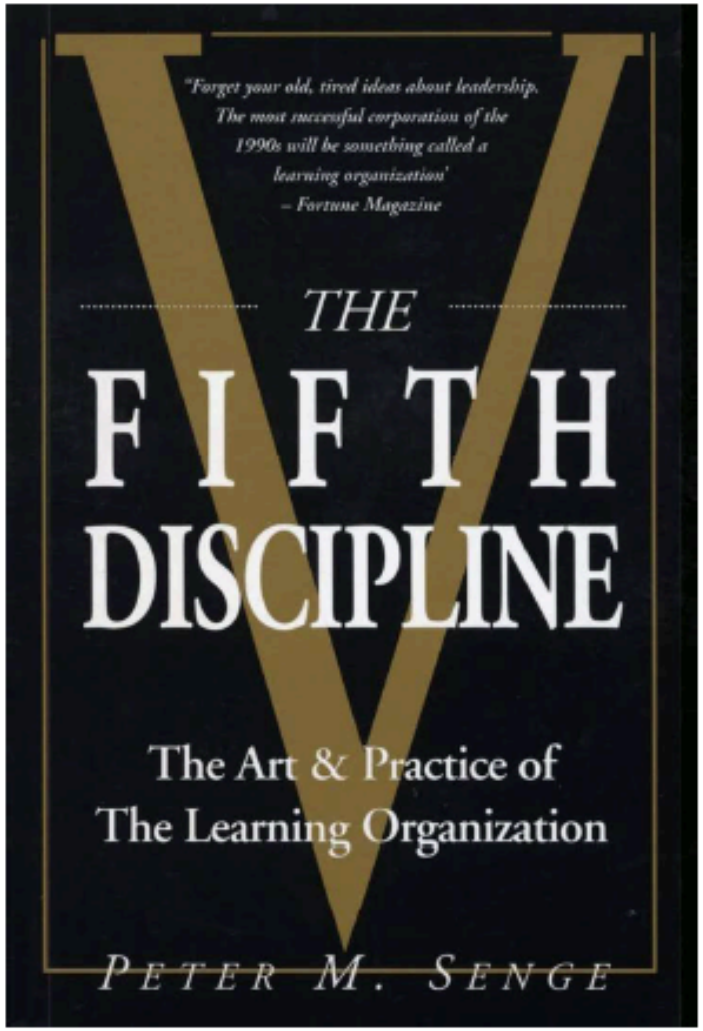
1. L'expérimentation continue, prendre des risques et apprendre des échecs
2. Comprendre que la répétition et la pratique sont les conditions préalables à la maîtrise

- Allouer du temps pour l'amélioration du travail quotidien
- Créer des rituels qui récompensent l'équipe de la prise de risques
- Introduire des défauts dans le système pour augmenter la résilience
- Prévoir du temps pour des expériences et des innovations en toute sécurité (hackathons)

La Formation & Expérimentation continue: Pratiques

Organisations apprenantes

- S’engager à apprendre
- S'améliorer nécessite d'apprendre quelque chose de nouveau
- Ne pas apprendre génère une dette culturelle
- Les humains aiment l’excellence (et aussi l'autonomie et le partage d’objectifs)
- L'engagement de la direction est essentiel



Encourager une culture d'apprentissage

- Encourager l'apprentissage quotidien et le partage des connaissances
- Créer des plans de formation et d'éducation basés sur les compétences
- Incorporer l'apprentissage dans les processus
- Utiliser la technologie pour accélérer l'apprentissage
- Faire en sorte que le travail soit éducatif par le biais d'expérimentations, de résolutions de problèmes et de démonstrations
- Autoriser et utiliser des erreurs comme sources d'apprentissage
- Rendre les résultats de l'apprentissage visibles

« Soit vous êtes une organisation en apprentissage, soit vous perdez au profit d'une qui l'est."

Andrew Shafer dans 'Beyond the Phoenix Project'

Ingénierie de la résilience

Capacité intrinsèque d'un système à adapter son fonctionnement avant, pendant ou après les changements et perturbations, afin de pouvoir poursuivre les opérations requises dans des conditions attendues ou inattendues.

- L'ingénierie de la résilience examine le fonctionnement de l'organisation dans son ensemble
- La meilleure défense est une bonne attaque
- Adopter une vue agressive, irréprochable et systémique après un incident
- Considérer à la fois les éléments humains et techniques
- Les systèmes doivent être plus forts que leur maillon le plus faible

"L'échec est le côté pile du succès" Eric Hollnagel

Préparation à l'echec

l'échec est une **situation courante** en expérimentation en tant que validation d'hypothèses.

l'échec devient moteur lorsqu'il est **dédramatisé, reconnu et mesuré** apporte au min. une information i.e, l'invalidation d'une hypothèse

l'échec est moins pénalisant si des **plages d'expérimentation** sont allouées dans les plannings

l'échec peut révéler une confusion ou une incompréhension
il faut donc **le communiquer** l'environnement projet

Improvement Kata

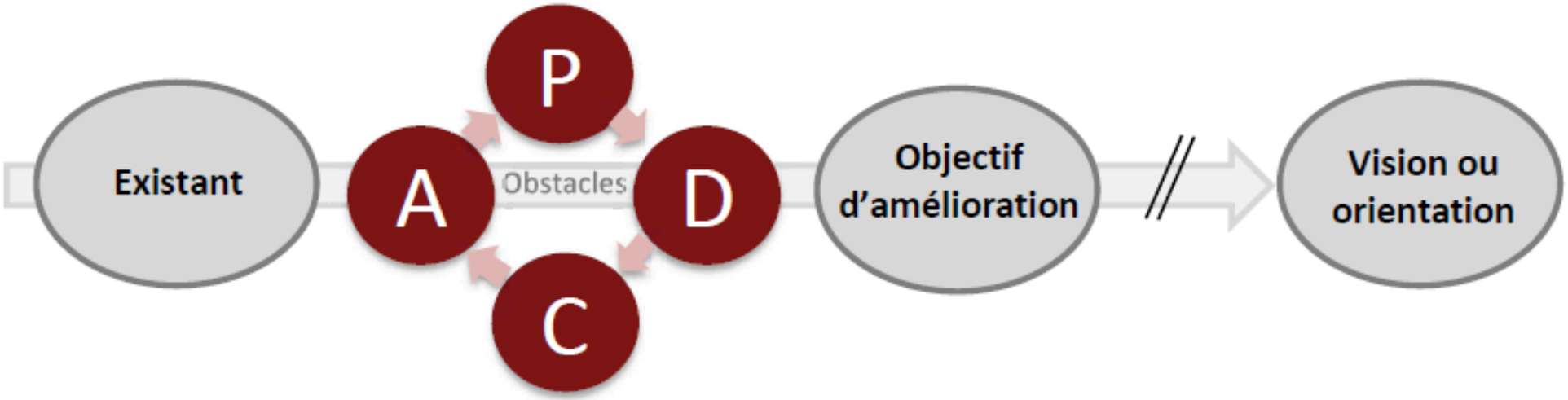
Un kata est une manière structurée de penser et d'agir que vous pratiquez jusqu'à ce qu'elle devienne une habitude.

2 Analyser l'existant

4 PDCA / expérimenter vers l'objectif

3 Définir un objectif d'amélioration

1 Comprendre la vision ou l'orientation à long terme



Le Improvement Kata est un processus en quatre étapes qui met l'accent sur l'apprentissage et l'amélioration du travail. Il prend en compte la vision ou l'orientation à long terme de l'organisation.
Plan > Do > Check > Act (PDCA)

Chaos Engineering

- Le concept « Simian Army » a été adopté pour la première fois par Netflix comme un service mettant fin de manière aléatoire à une instance de production.
- La réponse aux attaques permet de créer des compétences pour restaurer l'environnement de production suite à des défaillances inévitables.

« Chaos Monkey est un robot qui désactive de manière aléatoire nos instances de production afin de nous assurer que nous puissions surmonter ce type de défaillance ordinaire sans aucun impact sur les clients. Le nom vient de l'idée de libérer un singe sauvage avec une arme dans votre centre de données (ou région de cloud) qui casserait les instances et rongerait des câbles au hasard - tout en continuant de servir nos clients sans interruption. En activant les Chaos Monkeys pendant les jours ouvrables dans un environnement monitoré avec soin par des ingénieurs prêts à résoudre tout problème, nous pouvons toujours tirer les leçons des faiblesses de notre système et mettre en place des mécanismes de reprise automatique.
Ainsi, la prochaine fois qu'une instance échouera à 3 heures du matin le dimanche, nous ne le remarquerons même pas. »
Netflix

Ingenierie du Chaos

Les systèmes informatiques modernes sont *complexes*: interactions entre services implémentés sur des infrastructures distribuées.

Les différents avatars (pannes, incidents réseaux, charges...) provenant du système ou de l'environnement, associés à cette complexité rendent imprévisibles la réponse du système, ce qui est la *signature d'un phénomène chaotique*.

> **L'ingénierie du chaos est un mode d'expérimentation contrôlée visant à mesurer la confiance dans l'exploitation d'un produit soumis à des turbulences dans son environnement.**

Ingenierie du chaos: processus

1. Mesurer la réponse du système en **régime permanent**
2. Introduire des évènements impromptus **« chaos variables »** :
   - crash serveur / stockage / réseau
   - dysfonctionnement d'un service dans le système
3. Mesurer **l'écart** de la réponse du système soumis à ces variables, au régime permanent
4. **Automatiser** l'éxpérience pour pouvoir la mener en continu
5. Si possible, Mener l'expérience sur **l'env. de prod**, de façon contrôlée *déploiement bleu-vert, canari*