PeopleCert DevOps « Les fondamentaux »

Module 4 :
Le Full Stack – Processus et pratiques

Sujets :

- Evolution des pratiques DevOps
- Les 15 pratiques essentielles de DevOps

Module 4 : Le Full Stack – Processus et pratiques

Evolution des pratiques DevOps

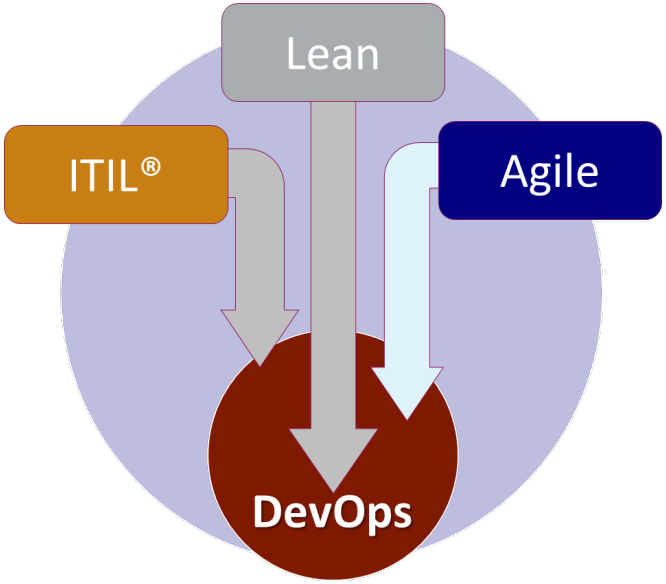
DevOps et les autres référentiels

- La plupart des pratiques utilisées par DevOps s'inscrivent parfaitement dans le cadre plus large de la gestion des services informatiques et trouvent leur origine dans une variété d'autres référentiels ITSM.

La **gestion des services** est un ensemble d'aptitudes organisationnelles spécialisées permettant de créer de la valeur pour les clients sous la forme de services.

Lors de la mise en œuvre de DevOps, il est plus facile et plus rapide d'adopter et d'adapter un référentiel existant que d'en créer ou d'en inventer un à partir de zéro. C'est pourquoi DevOps utilise des pratiques issues notamment d'ITIL, du Lean et de l'Agile.

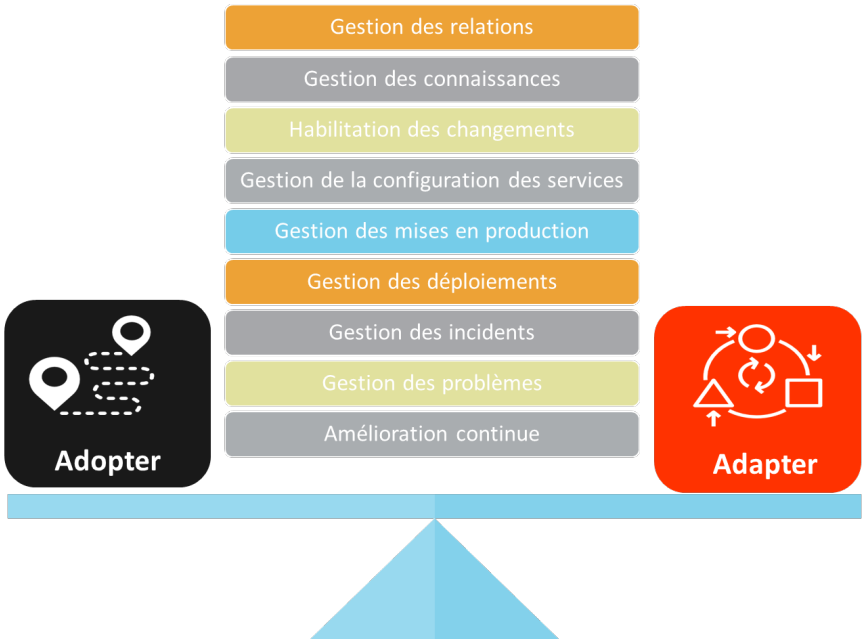
Ces référentiels ou approches sont basés sur un arbre logique commun qui stipule que pour améliorer quelque chose, il faut commencer par le définir. Une fois défini, il peut être stabilisé et contrôlé, ce qui permet de le mesurer et de le surveiller. Sur la base d'écarts de mesure, des opportunités d'amélioration sont identifiées et fournissent des informations sur la manière dont la capacité doit être redéfinie.



Source : *ITIL Foundation ITIL 4 Edition*

DevOps et les autres référentiels (ITIL)

ITIL est un guide de bonnes pratiques pour la gestion des services informatiques (ITSM).



ITIL est conçu pour être adopté et adapté et n'est pas prescriptif ni spécifique à un fournisseur.

ITIL n'est pas destiné à être adopté comme une solution universelle : il doit être adapté en fonction des services pris en charge et de la technologie informatique sous-jacente.

Adopter ITIL signifie s'engager dans une culture orientée vers les services et le client, dans le cadre d'un changement au niveau de l'organisation.

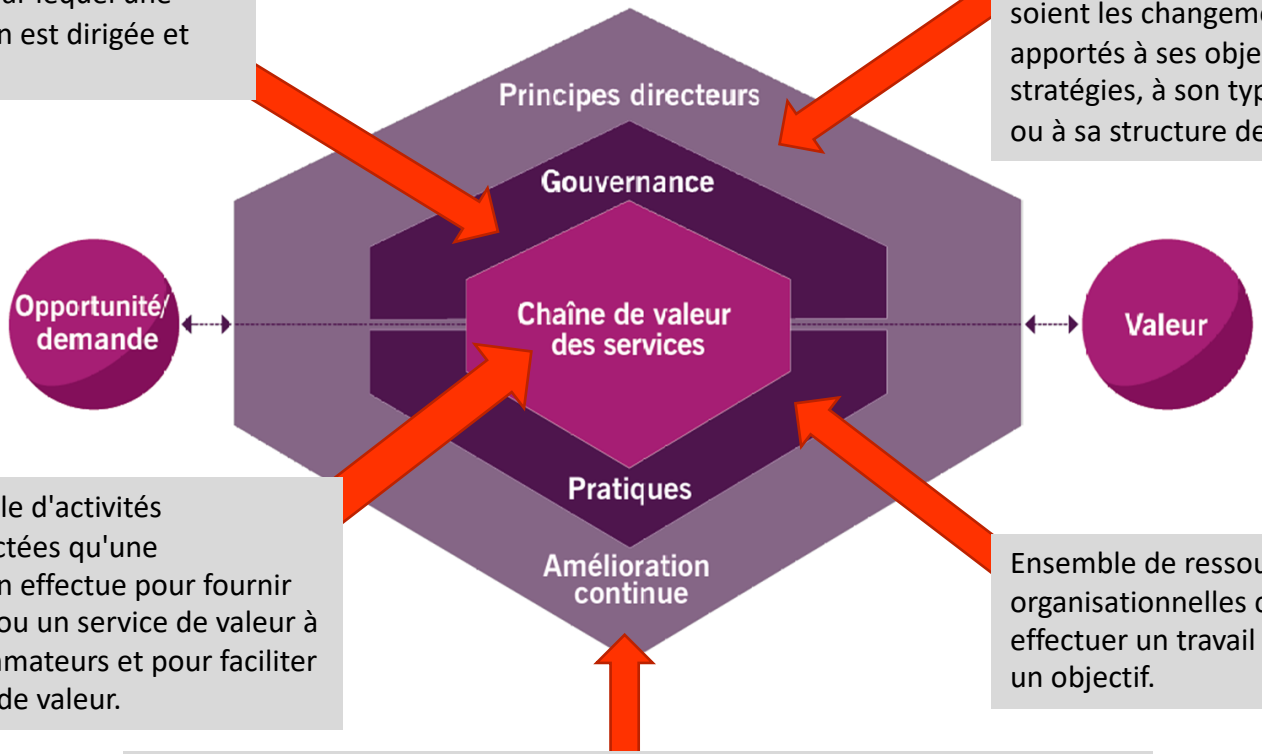
Adapter ITIL signifie comprendre les meilleures pratiques et les raisons pour lesquelles elles sont recommandées, puis les comparer attentivement aux besoins, buts et objectifs de l'entreprise.

Dans ce cours, plusieurs pratiques ITIL sont abordées et ont été adaptées pour DevOps.
Le Système de Valeur des Services (SVS) au cœur d'ITIL offre un moyen holistique de comprendre les opportunités et les demandes faites à l'informatique, les éléments ITSM et ce que l'informatique produit en sortie (la réalisation de valeur pour l'organisation, les clients et les parties prenantes).

DevOps et les autres référentiels (le SVS d'ITIL)

Le moyen par lequel une organisation est dirigée et contrôlée

Recommandations pouvant guider une organisation en toute circonstance, quels que soient les changements apportés à ses objectifs, à ses stratégies, à son type de travail ou à sa structure de gestion



Un ensemble d'activités interconnectées qu'une organisation effectue pour fournir un produit ou un service de valeur à ses consommateurs et pour faciliter la création de valeur.

Ensemble de ressources organisationnelles conçues pour effectuer un travail ou atteindre un objectif.

Une activité organisationnelle récurrente réalisée à tous les niveaux pour s'assurer que les performances d'une organisation répondent continuellement aux attentes des parties prenantes.

(*Figure 4.1 The ITIL Service Value System – ITIL® Foundation, ITIL 4 edition, 2019)*

DevOps et les autres référentiels (Lean)

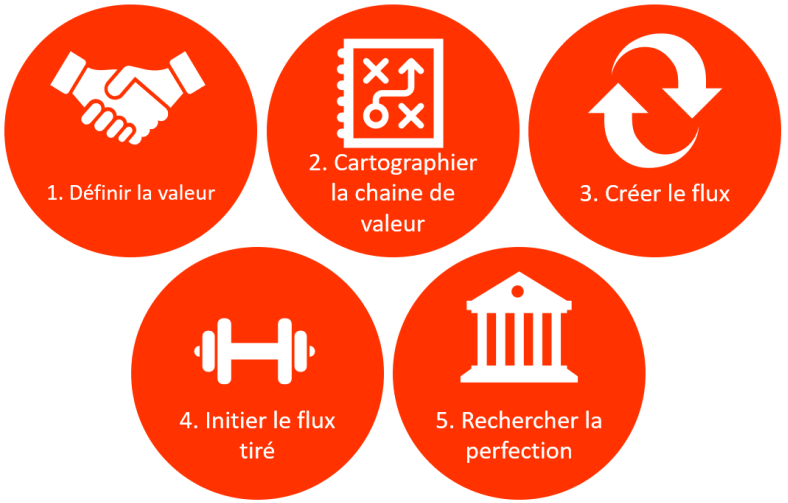
Lean est un système qualité axé sur le flux. Il s'agit avant tout d'accroître la valeur apportée au client, d'éliminer le gaspillage et d'améliorer en permanence.

Lean est né du Système de Production de Toyota (TPS), une série d'innovations mises en œuvre par Toyota après la Seconde Guerre mondiale. À cette époque, les objectifs étaient de mettre fin au ralentissement constant du travail et de remédier aux problèmes de qualité en étudiant chaque étape des processus, de gaspiller le moins possible les ressources disponibles et d'orienter le travail en fonction des ventes effectives plutôt que sur les objectifs de vente ou de production. Cela a augmenté la continuité et la vitesse de production, tout en permettant à Toyota d'offrir une gamme de produits plus large.

Aujourd'hui, le Lean est devenu un système essentiel dans le secteur industriel, et il a également fait ses preuves dans le secteur des services et dans celui de l'informatique.

Lean ne consiste pas à faire plus avec moins. Il consiste à faire plus en faisant moins.

Principes de la pensée Lean



Source : *Lean Thinking,* James P. Womack & Daniel T. Jones

DevOps et les autres référentiels (le manifeste Agile)

Les valeurs Agile

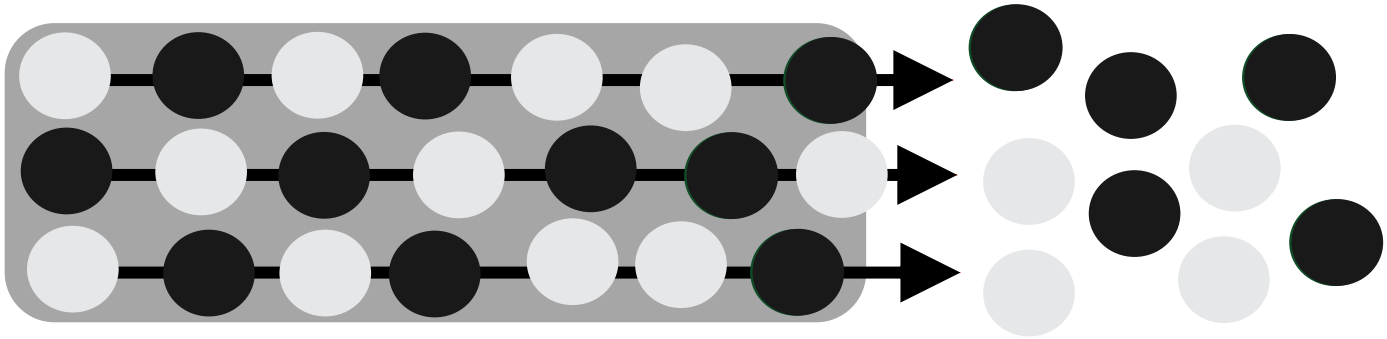
Les principes Agile

- Satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée
- Accueillez positivement les changements de besoins, même tard dans le projet
- Livrer fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois
- Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble
- Réaliser les projets avec des personnes motivées
- La méthode la plus simple et la plus efficace pour transmettre de l'information est le dialogue en face à face
- Un logiciel opérationnel est la principale mesure d'avancement
- Les processus Agiles encouragent un rythme de développement soutenable
- Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité
- La simplicité est essentielle
- Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées,
- À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace

Source : *The Agile Manifesto*, http://agilemanifesto.org

DevOps et les autres référentiels

- **DevOps est la solution qui fonctionne**. Il utilise la pensée et les pratiques Lean, ainsi que des approches/cadres pour la gestion de projet Agile tel que Scrum, et les approfondit pour s'attaquer au mur de la confusion (cette barrière entre Dev et Ops qui les empêche de se déplacer à la même vitesse et dans la même direction) et au goulot d'étranglement empêchant l'informatique de générer de la valeur.



Module 4 : Le Full Stack – Processus et pratiques

Les 15 pratiques essentielles de DevOps

Les 15 pratiques essentielles de DevOps

1. S'aligner sur la valeur métier en définissant la **Voix du client**
2. Accroître la collaboration et supprimer les silos avec la **Gestion des relations**
3. Identifier les contraintes et les goulots d'étranglement avec la **Cartographie de la chaîne de valeur**
4. Éliminer le gaspillage dans la mesure du possible et réduire les transferts avec **l'Optimisation des processus Lean**
5. Permettre le partage des connaissances et créer une culture d'apprentissage avec la **Gestion des connaissances**
6. Créer de la transparence dans le flux de valeur et rendre le travail visible avec le **Management visuel**
7. Travailler par petits lots et créer des boucles de feedback avec **Scrum agile**
8. Détecter les erreurs dès qu'ils se produisent et échouer plus rapidement avec le **Shift Left**
9. Trouver le juste équilibre entre flexibilité et stabilité avec l'**Habilitation des changements**
10. Intégrer les informations et les ressources là où vous en avez besoin, quand vous en avez besoin avec la **Gestion de la configuration des services**
11. Automatiser le support et créer du flux avec la **Gestion des mises en production**
12. Utiliser le « swarming » (mobilisation collective et cohérente) pour résoudre les incidents avec la **Gestion des incidents**
13. Transformer les connaissances en améliorations qui rapprochent la qualité de la source avec la **Gestion des problèmes** et Kaizen
14. Institutionnaliser les améliorations quotidiennes avec des modèles **d'Amélioration continue**
15. Reconnaître l'échec en tant qu'occasion d'apprentissage, en mettant l'accent sur **l'Antifragilité**

1. Voix du client (VOC)

Comprendre l'utilité et la garantie

- **L'utilité** représente les exigences fonctionnelles. Elle décrit les exigences d'un service qui sont adaptées à son usage prévu (le service fait-il ce qu'il est censé faire ?)
- La performance du service est-elle prise en charge ? Existe-t-il des contraintes empêchant son fonctionnement ? L'utilité est la fonctionnalité du service à répondre à un besoin spécifique.

La **garantie** représente des exigences non fonctionnelles. Elle décrit les exigences d'un service qui sont « adaptées à l'utilisation » - le service est-il réellement fourni ?

La garantie signifie que le service est disponible et fonctionne lorsque le client en a besoin, avec une capacité appropriée, de manière continue et sécurisée.

1. Voix du client (suite)

Voix du client Lean

La voix du client est outil majeur du Lean, qui permet de comprendre le client et ce qu'il définit exactement comme étant la valeur.

Lean propose de prendre en compte deux types de clients : ceux qui utilisent réellement le produit ou service et ceux qui sont juste derrière dans la chaîne de valeur.

Il est important de prendre en compte les quatre questions suivantes lors de la définition de la valeur dans Lean :

- Qui est le client ?
- Que veut le client ?
- Qu'est-ce qui ajoute de la valeur au client ?
- Combien le client est-il prêt à payer ?

Comprendre la notion de « Critique pour la Qualité » (CTQ)

Critique pour la Qualité sont des éléments indispensables. Selon les principes de Lean, ce sont ces éléments Essentiels pour la Qualité qui doivent être priorisés et ciblés. Si vous ne les livrez pas, vous ne fournissez pas vraiment de valeur.

2. Gestion des relations

La gestion des relations vise à établir et à maintenir la relation entre l'informatique et les parties prenantes, à identifier la valeur métier et à s'assurer que l'informatique la comprend et peut la fournir.

La personne qui rend réellement efficace le processus de gestion des relations en agissant comme intermédiaire entre l'informatique et les partenaires métiers internes.

Le rôle du Relationship Manager est généralement à temps plein, car il est essentiel pour entretenir les relations nécessaires et garantir que les services fournis par le service informatique répondent aux besoins et apportent la valeur métier nécessaire.

Ce rôle est parfois perçu comme créant des silos car on peut penser qu'il permet aux entreprises et aux services informatiques d'éviter la communication. Rien n'est plus loin de la vérité. Le Relationship Manager peut et doit être un rôle qui permet la collaboration en agissant moins en tant qu'intermédiaire et plus en tant que facilitateur, en faisant les présentations et en s'assurant que les deux côtés se comprennent.

3. Cartographie de la chaîne de valeur

Qu'est ce qu'un processus ?

Un **processus** est un ensemble structuré d'activités conçues pour atteindre un objectif spécifique. Il prend une ou plusieurs entrées définies et les transforme en sorties définies.

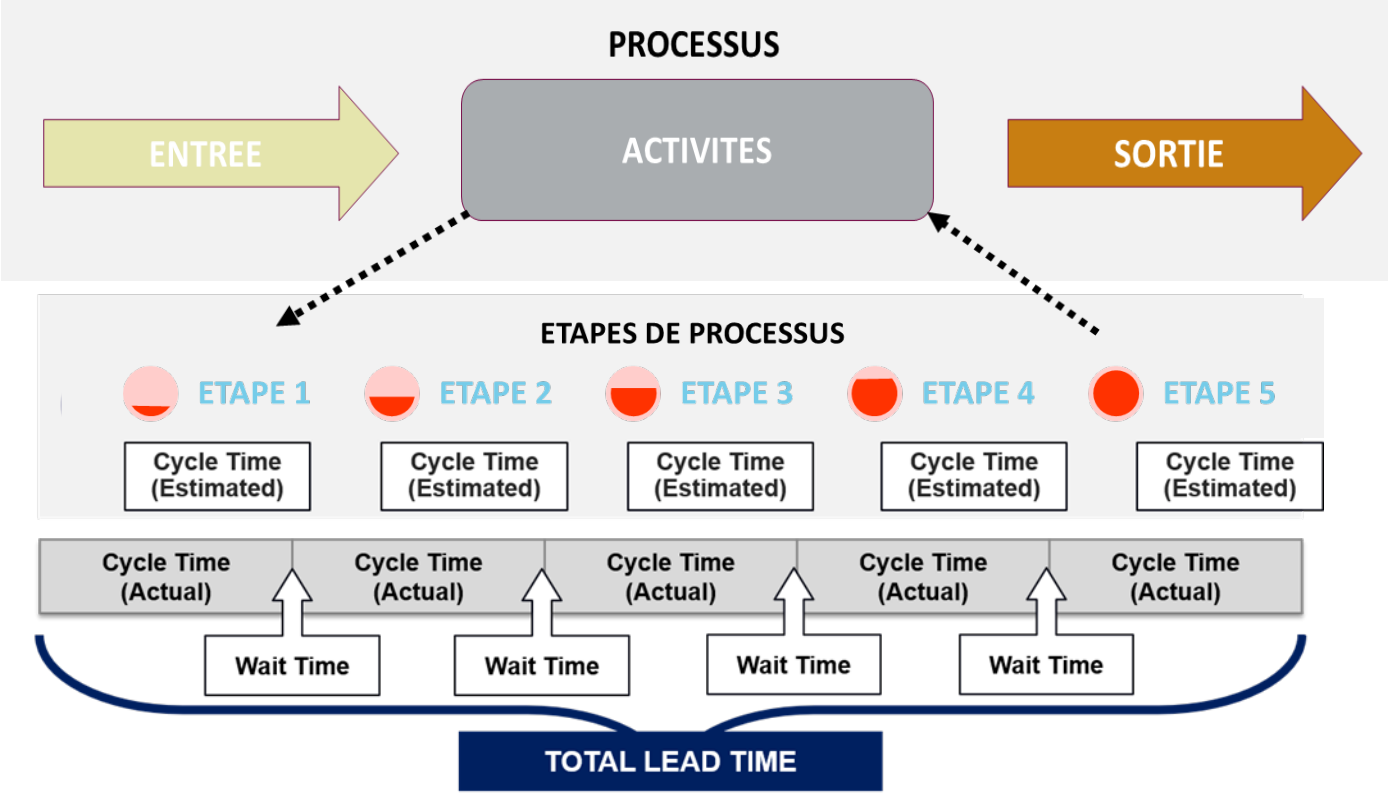
La chaîne de valeur

La chaine de valeur fournit une vision holistique de la façon dont l'informatique apporte de la valeur au métier. Comprendre non seulement où se situe votre place dans la chaîne de valeur, mais la chaîne de valeur dans son ensemble est essentiel pour le leadership et la gouvernance.

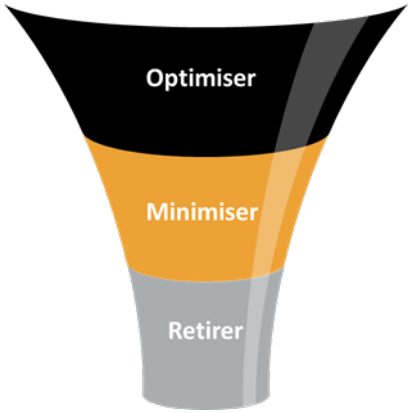
L'un des principes clés de Lean est de cartographier la chaîne de valeur - cela signifie mesures et métriques, en utilisant des données pour se concentrer sur des problèmes particuliers ou pour déterminer où a lieu le gaspillage et pourquoi il a lieu, afin de pouvoir ensuite établir des priorités de manière efficace et définir les zones où une amélioration rapide est possible.

3. Cartographie de la chaîne de valeur (Suite)

- **Le temps de cycle (cycle time)** est le temps séparant deux étapes/événements successifs dans un même processus.
- **Le temps de traitement (process time)** est le temps nécessaire à une ou plusieurs entrées pour être transformé en produit ou service fini.
- **Le temps d'attente (wait time)** est le temps passé à attendre que les prochaines étapes commencent.
- **Le lead time** correspond au temps écoulé depuis l'entrée dans le processus jusqu'à sa sortie - les temps de traitement d'attente sont additionnés.



4. Optimisation des processus Lean



- **L'activité à valeur ajoutée** doit être optimisé. Il s'agit des activités qui construisent de la valeur pour le client et que celui-ci peut percevoir et expérimenter. C'est ce que le client est prêt à acheter.
- **L'activité à non valeur ajoutée nécessaire** doit être minimisé. C'est une activité qui n'apporte pas directement de valeur ajoutée mais qui est nécessaire.
- **Le travail à non valeur ajoutée** doit être supprimé.

Gaspillage Lean : "Muda"

Transport
Inventaire
Mouvement
Temps d'attente
Surproduction
Surtraitement
Défaut et correction
Compétence non utilisée

Autres formes de perte de valeur

"Mura"
Ecarts & variations
Les exemples incluent : la variabilité en volume, la propagation dans le résultat des processus

"Muri"
Surcharge et inflexibilité
Les exemples incluent : impossibilité d'augmenter ou de diminuer la demande, délais de service fixe

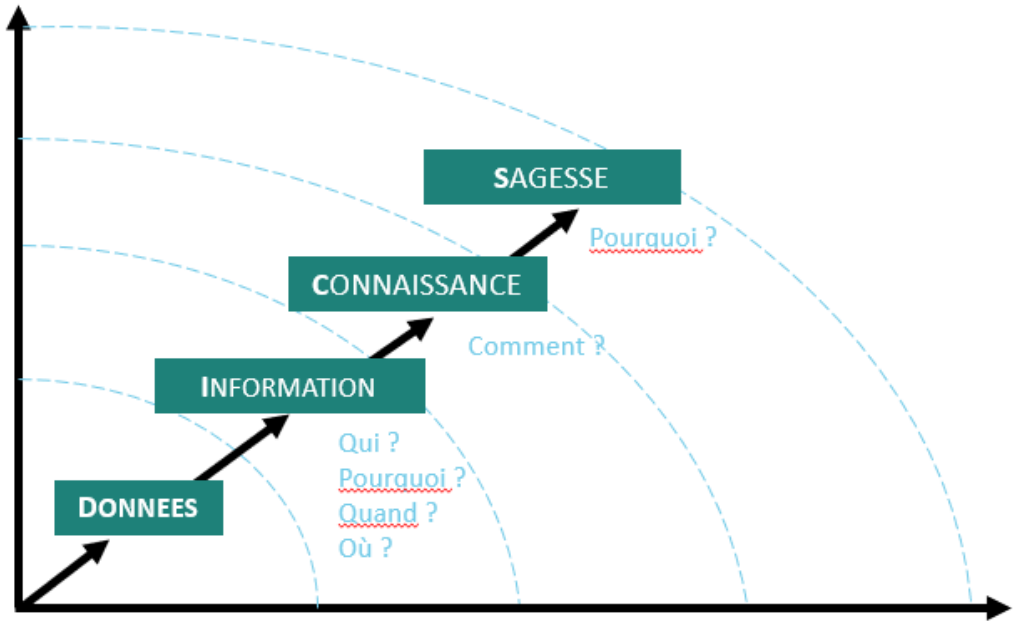
5. Gestion des connaissances

La gestion des connaissances consiste à intégrer et à rendre accessibles les connaissances essentielles requises pour prendre en charge un environnement DevOps et garantir des services stables et efficaces.

- La gestion des connaissances vise à :
  - Partager des perspectives, des idées, des expériences et des informations
  - S'assurer qu'elles sont disponibles au bon endroit, au bon moment, pour permettre des décisions éclairées
  - Améliorer l'efficacité en réduisant le besoin de redécouvrir les connaissances

Le modèle DIKW (Data, Information, Knowledge and Wisdom)

Le **modèle DIKW** nous aide à comprendre la différence entre un manque de données ou d'informations et un manque de connaissances ou de sagesse.



5. Gestion des connaissances (suite)

- **Les données** sont des faits objectifs, des observations, des signes ou des symboles. Elles n'ont ni contexte ni signification et ne sont utilisables que si le contexte est appliqué.
- **Les informations** sont des données organisées ou structurées qui ont été définies dans un contexte ou dans un but.
- **La connaissance** est une information encadrée par des expériences, des valeurs, une vision ou une intuition.
- **La sagesse** nécessite des contextes de niveau supérieur et une pensée relationnelle.

Exemples de systèmes de gestion des connaissances

Technologies logicielles collaboratives, telles que Slack ou un wiki interne	Bases de connaissances, tel qu'un système de gestion documentaire
Mentorat et observation au poste de travail	Intégration de compétences variées dans des équipes pluridisciplinaires
Réunions informelles telles que des rencontres du savoir pour échanger des idées ou enseigner aux autres	Outils de management visuel tel que Kanban

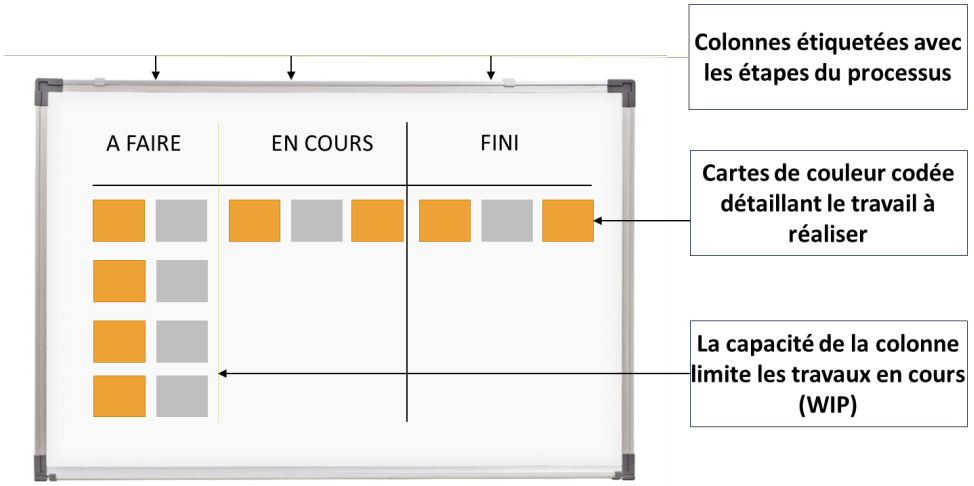
6. Management visuel

Le management visuel est un outil permettant de simplifier au maximum le contrôle et la gestion d'une organisation. La pratique consiste à utiliser des contrôles visuels plutôt que basés sur du texte, à partir desquels il est plus facile d'interpréter, de se souvenir et de garder en tête.

- Le management visuel a deux objectifs :
  - Augmenter l'efficience et l'efficacité d'un processus et d'une chaine de valeur en rendant les étapes visibles.
  - Rendre les problèmes, les anomalies ou les écarts par rapport aux normes visibles pour tout le monde. Lorsque ces écarts sont visibles et apparents à tous, des mesures correctives peuvent être prises pour corriger immédiatement ces problèmes.

Utiliser Kanban

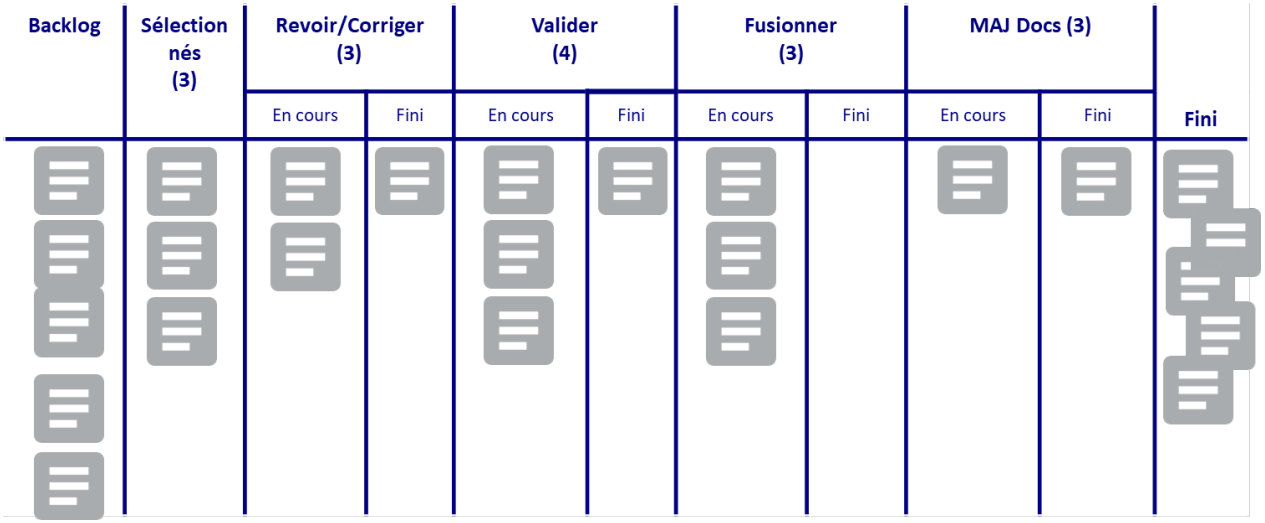
Kanban est apparu dans les années 1940 dans le cadre de l'évolution initiale de la fabrication sans gaspillage. Il permettait aux travailleurs de la chaîne de montage d'informer leurs partenaires en aval de la demande de pièces ou d'autres tâches. Il a permis une transparence et une communication accrue, ainsi que des processus normalisés.



6. Management visuel (suite)

Systèmes à flux poussé par rapport à flux tiré et limites des travaux en cours (WIP)

Dans un **système à flux tiré**, les produits ou les services sont tirés à travers le processus par la demande du client, contrairement à un **système à flux poussé** dans lequel des produits ou des services sont poussés tout au long du processus en fonction de la demande prévisionnelle. Les systèmes à flux tiré réduisent le gaspillage dans les processus.



Les limites des travaux en cours sont des maximums ou des minimums spécifiques pour l'ajout de nouveaux travaux qui garantissent que la quantité de travail ne dépasse jamais la capacité du système à le réaliser.

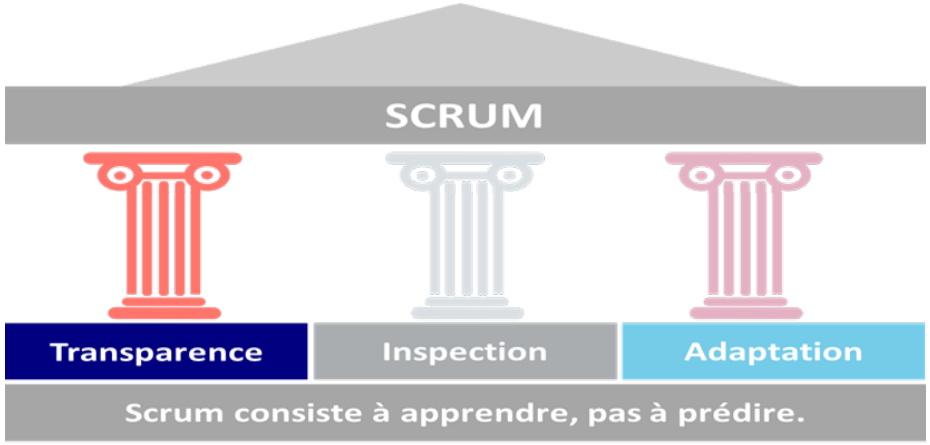
7. Scrum Agile

Comprendre Scrum

Alors que pour beaucoup, Agile et Scrum sont maintenant synonymes, Scrum est en fait utilisé depuis le début des années 90 pour gérer le développement de produits complexes. C'est un cadre de processus adaptable dans lequel divers processus, outils et techniques peuvent être appliqués. Il favorise le développement de produits de manière itérative, ce qui entraîne des versions plus fréquentes avec des résultats de la plus haute qualité possible.

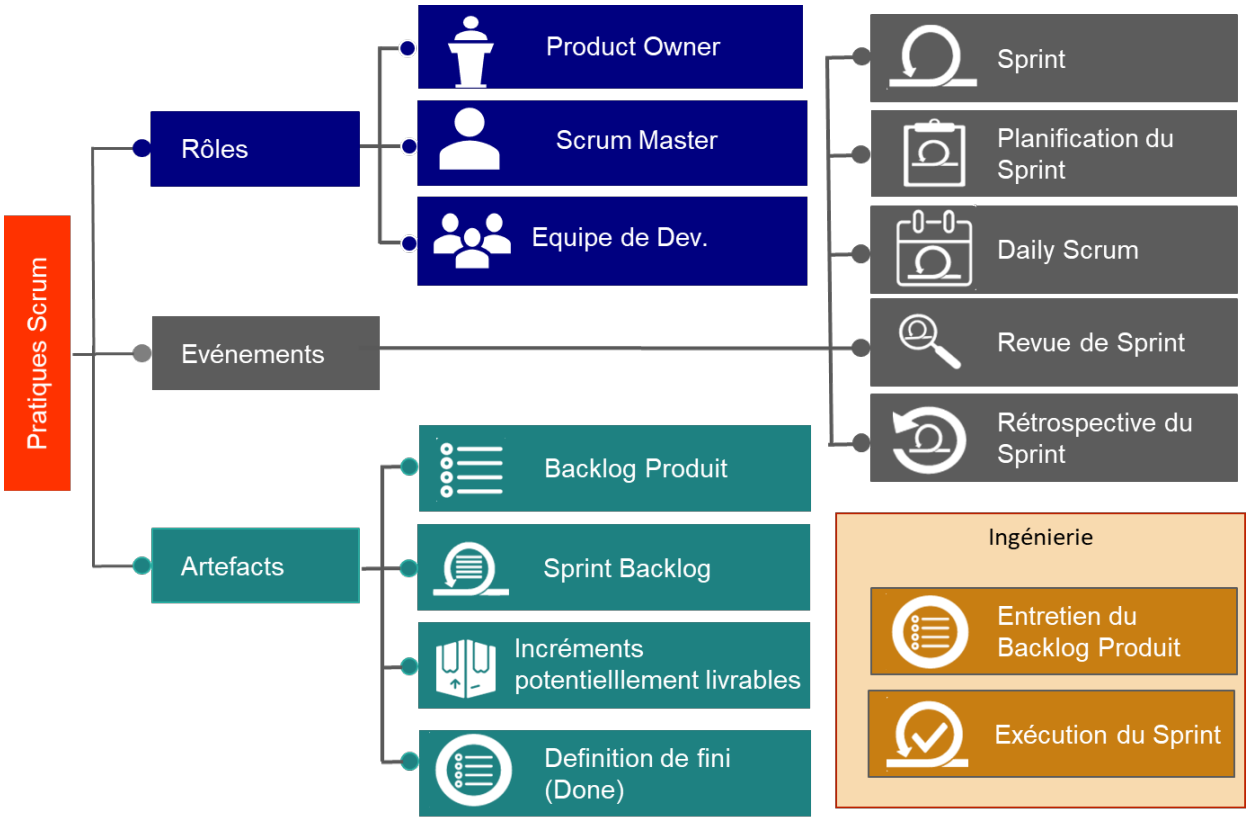
Scrum repose sur trois piliers essentiels.

- **Inspection**. Dans Scrum, des inspections de la progression et des artefacts sont effectués fréquemment et de manière cohérente tout au long du projet. Un retour d'informations est alors recueilli pour garantir que tout est toujours sur la bonne voie.
- **Transparence**. Cela signifie rendre visible le processus autant que possible aux responsables du résultat et faire en sorte que tout le monde s'accorde sur une définition standard du terme « Fini ».
- **Adaptation**. Lorsqu'une inspection est effectuée, si des aspects d'un processus entraînent des défauts, des erreurs ou une qualité réduite, des ajustements sont effectués immédiatement. Des événements formels assurent cela.



7. Scrum Agile (suite)

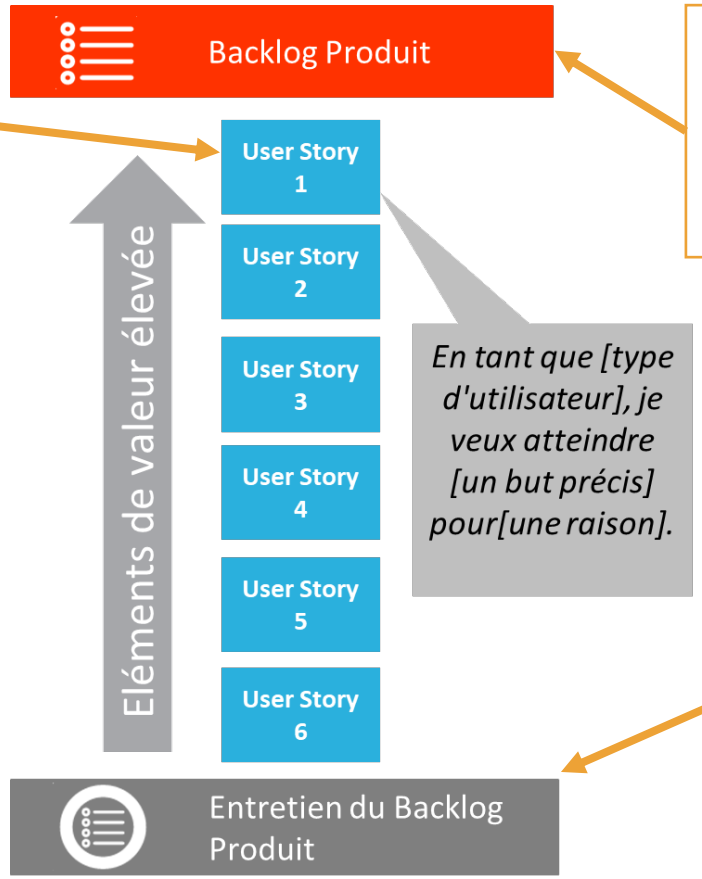
Scrum en un coup d'œil



7. Scrum Agile (suite)

Le Backlog Produit

Les user stories sont des fonctionnalités de produit ou des corrections de bugs qui sont examinées du point de vue de l'utilisateur final.



Le Backlog Produit est constitué de toutes les demandes de modification et classées par ordre de priorité en fonction de leur importance.

En tant que [type d'utilisateur], je veux atteindre [un but précis] pour[une raison].

Le raffinement du Backlog Produit est facilité par le product owner avec la contribution du client et d'autres parties prenantes clés. Ceci est réalisé en permanence.
Les items en haut de la pile sont les plus importants et il sont suffisamment précis et réalisables à l'intérieur d'un sprint.
Les items en bas de la pille ne sont pas importants et peuvent rester flous et grossiers.

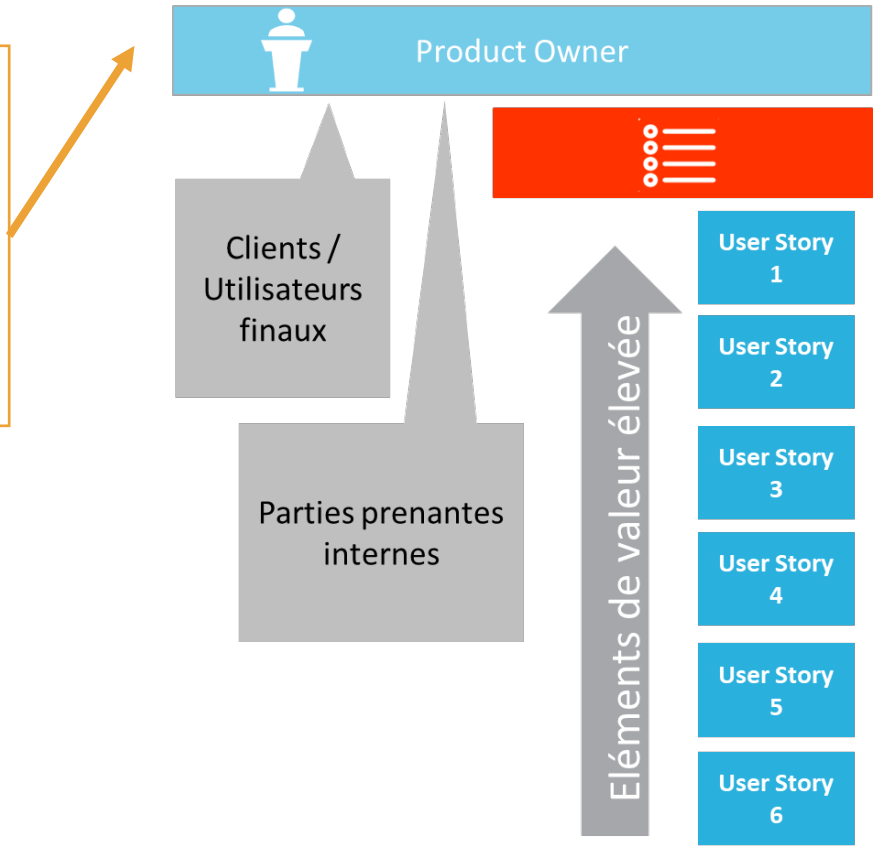
7. Scrum Agile (suite)

Le Product Owner

Le **Product Owner** est un optimiseur et un maximiseur de la valeur du produit et du travail de l'équipe.

Il prend ses décisions dans ce sens. Et ses décisions se traduisent dans le contenu et la priorisation du Product Backlog.

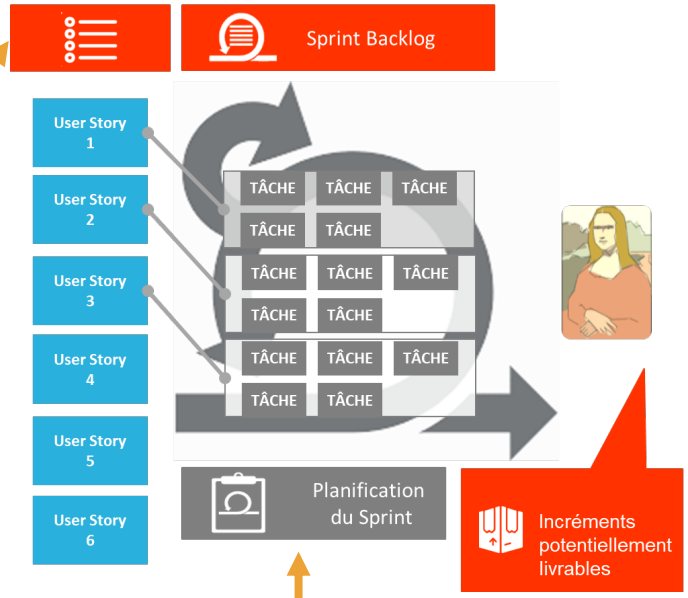
Il est en communication constante avec le client et collabore avec l'équipe.



7. Scrum Agile (suite)

Le Sprint Backlog

Le **Sprint Backlog** est le plan des travaux à réaliser dans une itération. Il contient le « what », une sélection des items du Product, le « how », la liste des tâches à engager et le « why », le Sprint Goal



Les incréments potentiellement livrables sont les éléments complétés sur la base du Sprint Backlog selon une « Définition de fini » convenue et des critères d'acceptation relatifs aux exigences fonctionnelles et non fonctionnelles.

La réunion de **Planification du Sprint** a lieu entre le Product Owner et l'équipe de développement. Le Product Owner présente son intention de ce qui devrait être réalisé à la fin du sprint. L'équipe de développement sélectionne les items du Backlog Produit qui doivent être réalisés et les deux acteurs négocient jusqu'à obtenir un Sprint Goal finement défini et « signé » par les deux parties.

L'équipe de développement est ainsi responsable du Sprint Backlog et s'engage sur l'atteinte de l'objectif négocié.

7. Scrum Agile (suite)

Le Scrum Master et l'équipe de développement

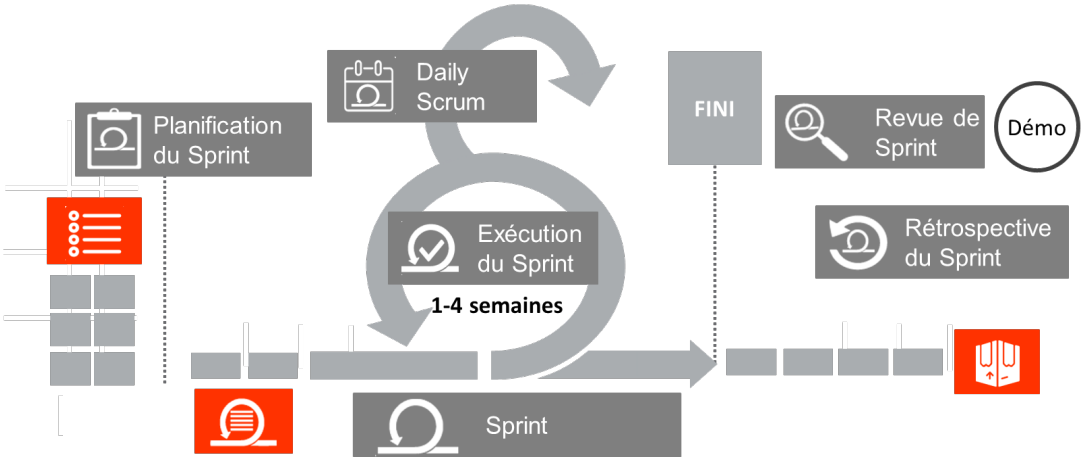
Le **Scrum Master** s'assure que le cadre Scrum est suivi complètement et correctement. Il a un rôle de facilitateur en tant que coach et est formé et capable de résoudre des problèmes. Il clarifie toutes les questions relatives à Agile et à Scrum et peut remplacer le Product owner si celui-ci est absent, pour assurer la maintenance du Backlog Produit.

L'équipe de Développement est un ensemble d'experts techniques auto-organisés et pluridisplinaires, capables de développer des solutions aux problèmes rencontrés. Ils possèdent et gèrent le Sprint Backlog et sont responsables de la création, des tests et de la mise en œuvre des éléments du Backlog Produit apportés.

7. Scrum Agile (suite)

Evénements Scrum et gestion par limitation de la durée (Timebox)

Dans Scrum, tous les événements ont une durée maximale. Cela signifie qu'ils sont effectués dans un délai maximum convenu auparavant. Lorsque le temps imparti est atteint, le travail est arrêté et les réalisations sont évaluées.

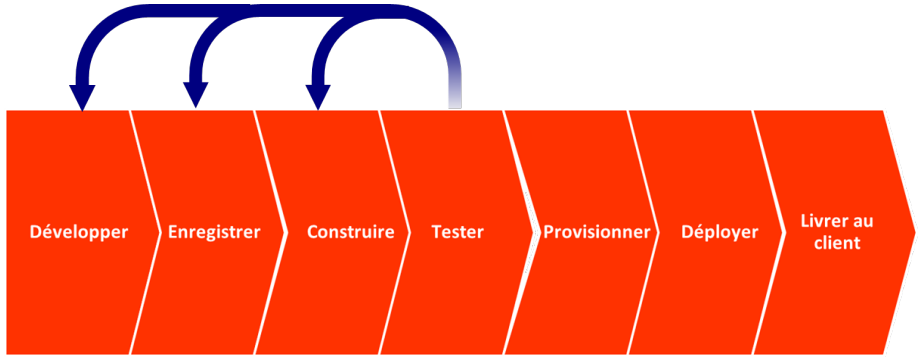


- Le **sprint** est un conteneur comprenant la planification du Sprint, l'exécution du Sprint, la revue de Sprint, et la rétrospective du Sprint, l'ingénierie et le raffinement.
- **L'exécution du sprint** commence après la planification du sprint et se termine lorsque le temps imparti est échu. Au cours de l'exécution, les éléments du sprint backlog deviennent des travaux en cours et sont traités jusqu'à leur achèvement conformément à la définition de Fini.
- Le **Daily Scrum** est une réunion conduite par l'équipe de développement pour avoir une vision partagée de l'avancement.
- La **revue de sprint** est tenue à la fin de l'exécution du sprint. Tous les éléments terminés conformément à la définition de « fini » sont présentés au Product owner et aux autres parties prenantes clés. Cela permet de passer en revue l'incrément et le produit, de le critiquer et d'avoir l'occasion d'évoquer tout sujet utile pour anticiper ou corriger le produit.
- La **rétrospective du sprint** permet à l'équipe de développement de passer en revue son propre fonctionnement et d'apporter des ajustements ou des perfectionnements (inspection et adaptation) à la manière dont elle conduit le sprint.

8. Shift Left

Shitf Left

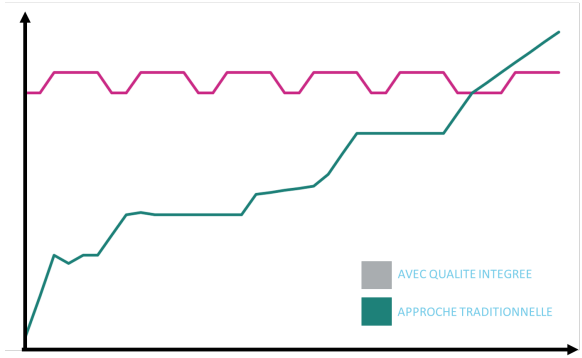
Shift Left garantit que la qualité est intégrée plus tôt dans le processus de développement afin que les problèmes soient détectés plus tôt et puissent être résolus, et que les défauts ou les erreurs n'affectent pas la production.



Faire passer la « Qualité en premier »

Le modèle « Qualité En Premier » préconise de limiter la portée et les fonctionnalités, mais offre plutôt un produit de haute qualité qui répond aux exigences non fonctionnelles, puis introduit davantage de fonctions et de fonctionnalités au fil du temps grâce à l'utilisation de sprints. Cela permet d'être plus souple avec l'ordre dans lequel les fonctionnalités sont livrées. Le modèle « Qualité En Premier » sous-promet et sur-livre plutôt que l'inverse.

Offrez TOUJOURS une haute qualité.



Source : *Practice To Perfect: The Quality First Model*

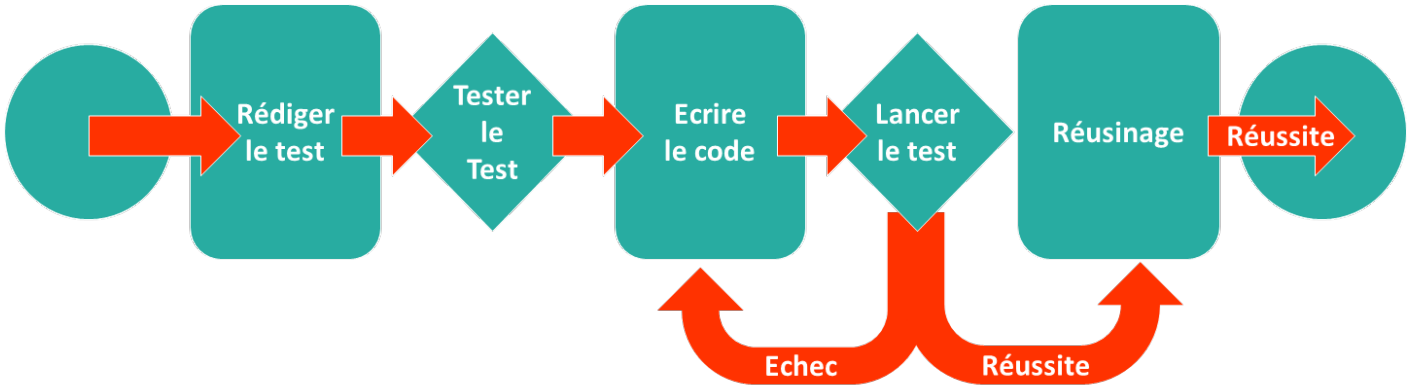
8. Shift Left (Suite)

Développement piloté par les tests

Le **développement piloté par les tests** (**Test Driven Development** - TDD) consiste à préparer des scénarios de test avant l'écriture d'un programme afin que le programmeur ait pour objectif d'écrire quelque chose qui puisse réussir le test exact.

Les avantages du TDD sont :

- les programmeurs savent exactement ce qu'on attend d'eux et peuvent travailler de manière plus productive ;
- les programmeurs sont encouragés à créer le programme le plus simple possible, rien que pour réussir le test. Ceci est souhaitable dans les environnements Agiles, car nous ne voulons pas perdre de temps à créer quelque chose de parfait, alors que tout ce dont nous avons besoin est d'une solution suffisamment bonne qui fasse le travail. Cependant, nous avons généralement besoin de garder un œil sur le réusinage pour nous assurer que la solution est toujours « suffisamment bonne ».



9. Habilitation des changements

L'habilitation **des changements** est une pratique ITIL qui contrôle le cycle de vie de tous les changements, ce qui permet d'apporter des modifications utiles avec un minimum de perturbation des services informatiques. Il optimise le risque global de l'entreprise.

Types de changement

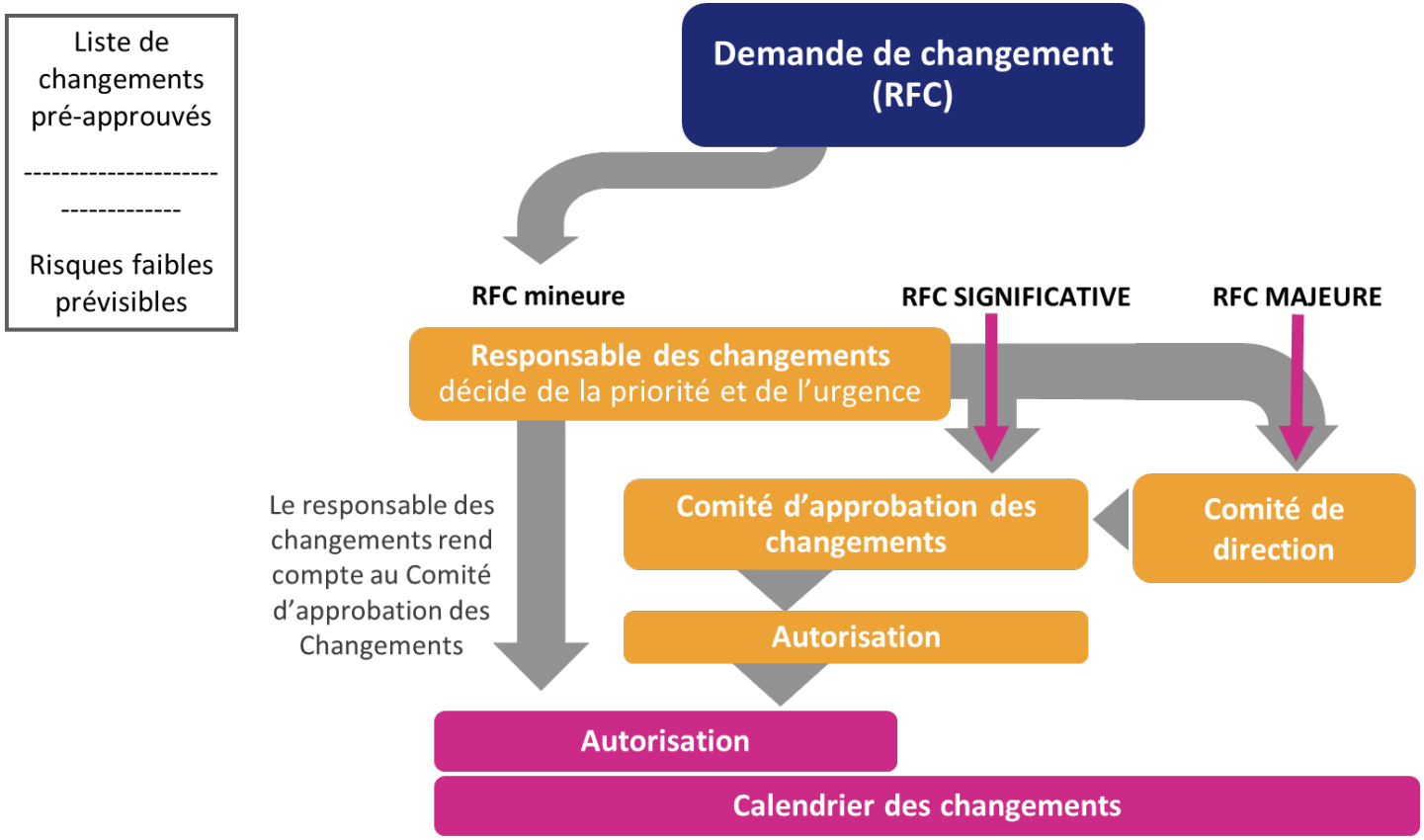
Les changements standards sont à faible risque, généralement courants et ont une procédure ou des instructions de travail spécifiques qui se sont avérés efficaces dans les mises en œuvre précédentes. Ces modifications peuvent être pré-autorisées.

Les changements urgents doivent être mis en œuvre immédiatement en réponse à un incident majeur ou à un besoin critique.

Les **changements normaux** sont des modifications qui ne sont ni standards ni urgentes.

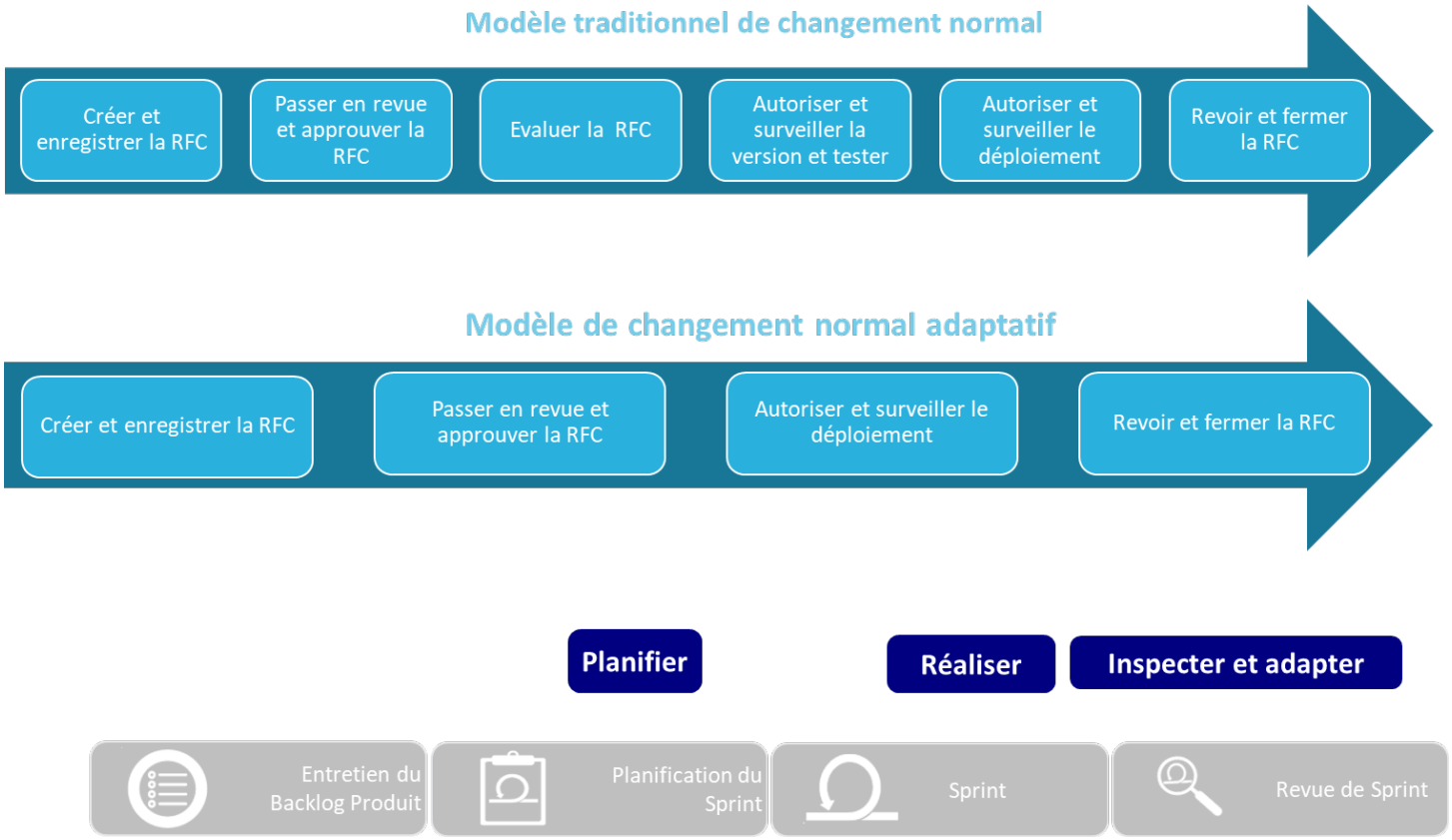
9. Habilitation des changements (suite)

Le processus d'approbation des changements standard



9. Habilitation des changements (suite)

Approbation des changements normaux pour DevOps



10. Gestion de la configuration des services

La gestion de la configuration des services est une pratique ITIL cruciale pour DevOps, car elle est essentielle pour l'automatisation. Ce processus consiste à maintenir la connaissance de l'état actuel de chaque composant de l'infrastructure.

- Dans un environnement DevOps, la gestion de la configuration de services englobe les pratiques et les outils qui automatisent la livraison et le fonctionnement de l'infrastructure. Les outils sont utilisés pour :
  - le modèle Infrastructure ;
  - appliquer en permanence les configurations souhaitées ;
  - Remédier aux changements inattendus ou la « dérive de configuration », les différences dans le temps entre les configurations d'infrastructure principale et secondaire.

Infrastructure en tant que Code (IAC)

La gestion automatisée de la configuration des services est réalisée dans DevOps en traitant l'infrastructure en tant que code pouvant être gérée avec les mêmes outils et processus utilisés par les développeurs, tels que le contrôle de version, la révision de code, les tests automatisés et les petits déploiements. L'Infrastructure en tant que Code repose sur plusieurs pratiques :

- utiliser des fichiers de définition et s'assurer que toute la configuration est définie dans des fichiers de définition exécutables ;
- mettre en œuvre des systèmes et des processus auto-documentés plutôt que des instructions à exécuter par les humains ;
- appliquer du versioning sur tout et garder tout le code dans votre contrôle de code source afin que chaque modification soit enregistrée ;
- tester en permanence les systèmes et les processus et trouver les erreurs rapidement dans la configuration de l'infrastructure ;
- mettre l'accent sur les petits changements plutôt que sur de gros lots, afin qu'il soit plus facile de trouver des erreurs et de revenir en arrière si nécessaire ;
- garder les services disponibles en permanence.

Source : https://martinfowler.com/bliki/InfrastructureAsCode.html

11. Gestion des mises en production

La Gestion des mises en production consiste à mettre à disposition des services et des fonctionnalités nouveaux ou modifiés. Elle comprend la gestion des versions et du déploiement.

La **version** désigne la création effective de services et de fonctionnalités nouveaux et modifiés.
Le **déploiement** fait référence au déplacement de composants de service nouveaux ou modifiés vers des environnements de production

Techniques de gestion des mises en production

Automatiser la gestion de la configuration des services avec l'infrastructure en tant que code (IAC)

La gestion de la configuration des services, automatisée et utilisant l'IAC, prend en charge votre processus de gestion des versions et contribue à accroître la fiabilité et à réduire les erreurs en automatisant le processus d'installation et de configuration du logiciel. Cela rend le processus de publication plus reproductible, car vous pouvez exécuter des scripts à plusieurs reprises dans les environnements de test et de production tout en maintenant le contrôle de version.

Reconstruire les serveurs régulièrement

Reconstruire régulièrement les serveurs à l'aide de machines virtuelles signifie pouvoir facilement détruire et recréer vos serveurs. Cela réduit l'impact de la dérive de configuration et vous oblige à vous assurer que votre code d'infrastructure contient des informations précises sur la configuration des serveurs.

Implémentation de déploiements sans interruption de service

L'implémentation de déploiements sans interruption de service implique de trouver des moyens de livrer votre logiciel sans affecter les utilisateurs finaux. C'est essentiel lorsque vous reconstruisez régulièrement vos serveurs, car sinon, le temps d'interruption requis serait inacceptable.

Source : https://techbeacon.com/3-devops-techniques-stress-free-release-management

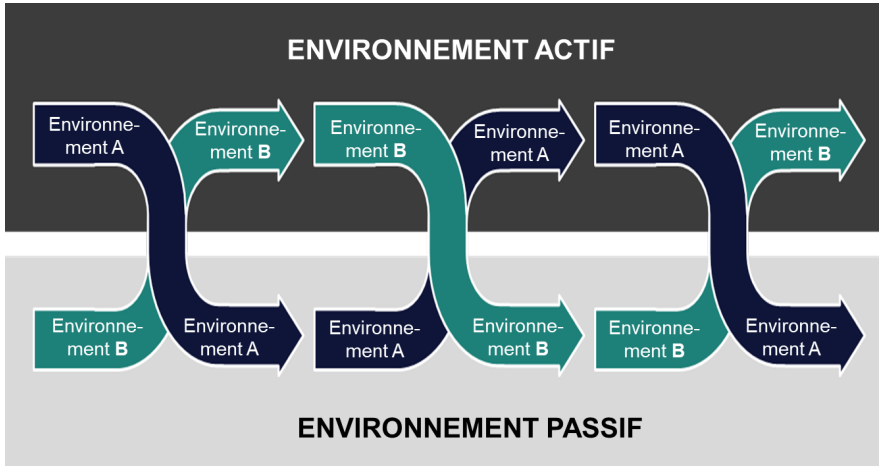
11. Gestion des mises en production (suite)

Déploiement Bleu-Vert

Le déploiement Bleu-Vert signifie la mise en place de deux serveurs de production identiques et la redirection de tout le trafic utilisateur allant vers un seul des ensembles de serveurs : un environnement « actif » en production. Cela vous laisse alors avec un environnement « passif » qui est utilisé uniquement pour le développement que les utilisateurs ne peuvent pas voir ou utiliser.

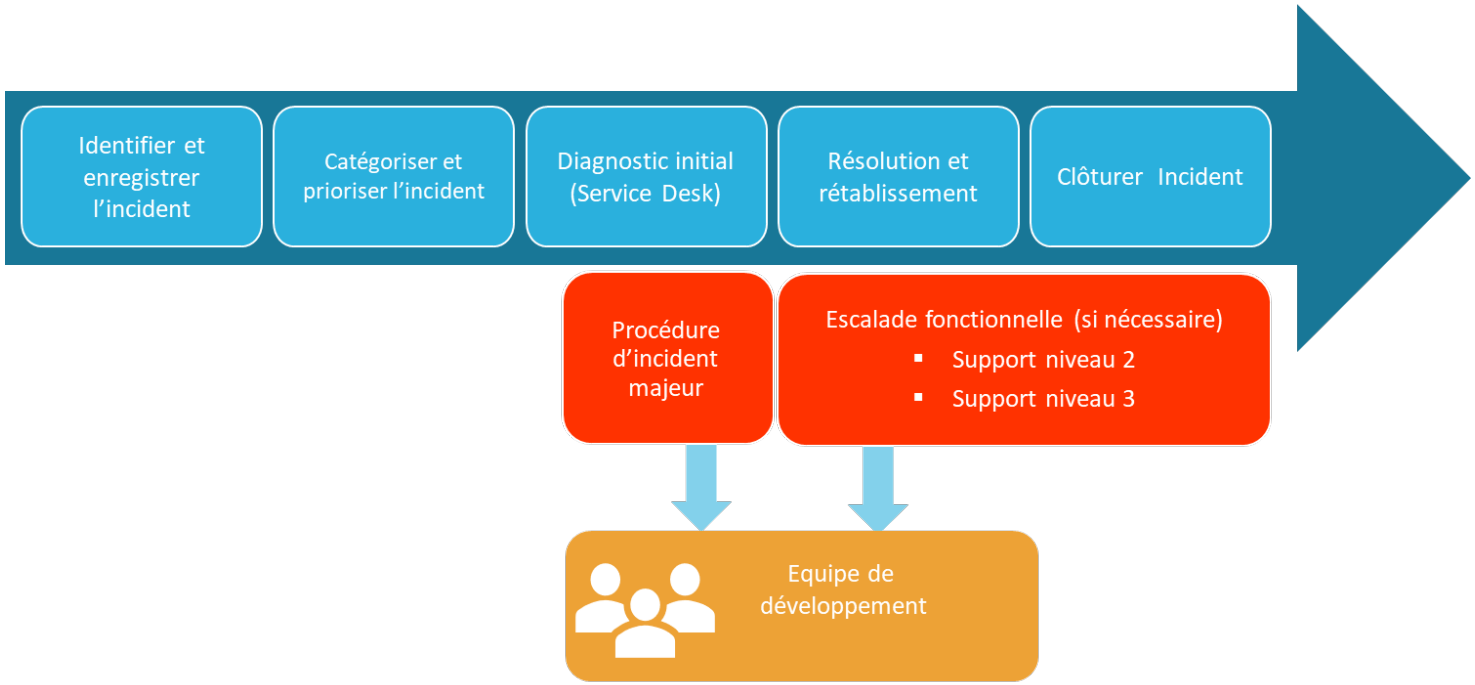
Avec le déploiement Bleu-Vert, vous pouvez déployer les modifications sur les serveurs que vous n'utilisez pas, les détruire et les reconstruire à votre guise, sans que personne ne soit affecté en cas de problème. Cela signifie que vous n'êtes plus obligé de déployer en dehors des heures normales afin de minimiser l'impact des temps d'arrêt ou de vous inquiéter de l'impact du changement sur les utilisateurs finaux.

Lorsque vous êtes certain que l'environnement passif fonctionne correctement, vous redirigez le trafic des utilisateurs vers le nouvel ensemble de serveurs. Ces serveurs deviennent le nouvel environnement actif et vous utilisez ensuite l'environnement précédemment actif comme environnement passif.



12. Gestion des incidents

La gestion des incidents est une pratique ITIL qui rétablit le fonctionnement normal du service le plus rapidement possible en cas d'incident, minimisant l'impact négatif sur les opérations métiers. Cela garantit le maintien des meilleurs niveaux possibles de qualité et de disponibilité du service.



12. Gestion des incidents (suite)

Incidents par rapport aux problèmes

- Il est important de noter la différence entre ce que ITIL qualifie d'**incident** et de **problème**. Selon ITIL, un incident est « une interruption imprévue d'un service, ou la défaillance d'un composant d'un service qui n'a pas encore eu d'impact sur le service ». Un problème, cependant, est la cause d'un ou de plusieurs incidents. Qu'un incident devienne étiqueté comme un problème est un choix dans ITIL basé sur une variété de considérations.

Incident

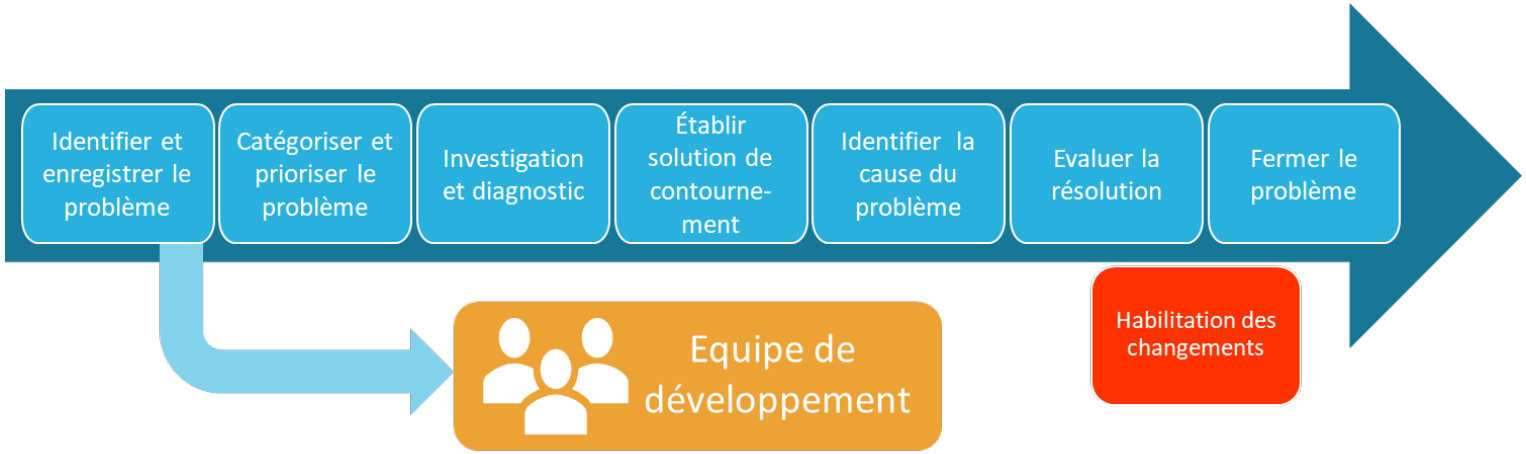
- Une interruption imprévue d'un service informatique ou une diminution de la qualité d'un service informatique
- Défaillance d'un élément de configuration qui n'a pas encore affecté le service

Problème

- La cause sous-jacente d'un ou plusieurs incidents
- La cause n'est généralement pas connue au moment de la création d'un enregistrement de problème, et le processus de gestion des problèmes est chargé de mener l'enquête.

13. Gestion des problèmes et Kaizen

La **gestion des problèmes** est une pratique ITIL qui vise à hiérarchiser, identifier et, si possible, résoudre la cause première d'un ou de plusieurs incidents afin de les éliminer ou de minimiser leur impact.



- Plus l'équipe de développement est responsabilisée lorsque quelque chose se casse, moins elle sera susceptible d'en arriver là. Cela diminue la mentalité de silo et encourage la collaboration et l'implication. Et, être impliqué dans des incidents et des problèmes permet également l'expérimentation et l'apprentissage qui sont au cœur du troisième principe.

13. Gestion des problèmes et Kaizen (suite)

Résolution de problème avec Kaizen

KAIZEN

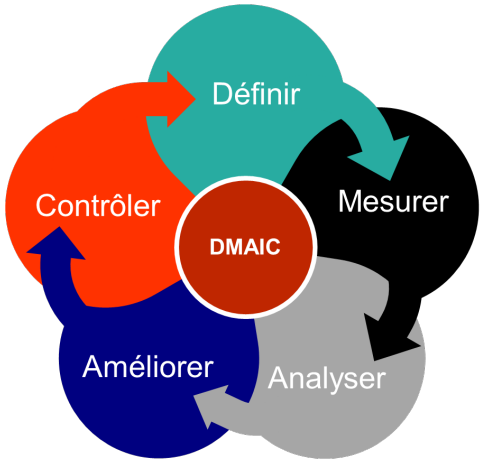
改善

CHANGER

POUR LE MIEUX

Kaizen est une approche structurée permettant de résoudre progressivement les problèmes et d'améliorer progressivement les processus et le flux, avec une attitude ou un état d'esprit encourageant tout le monde, à tous les niveaux de l'entreprise, à rechercher de petites idées qui, si possible, peuvent être mises en œuvre facilement et rapidement. Le Kaizen doit faire partie de la culture quotidienne d'une organisation.

Le cycle d'amélioration DMAIC



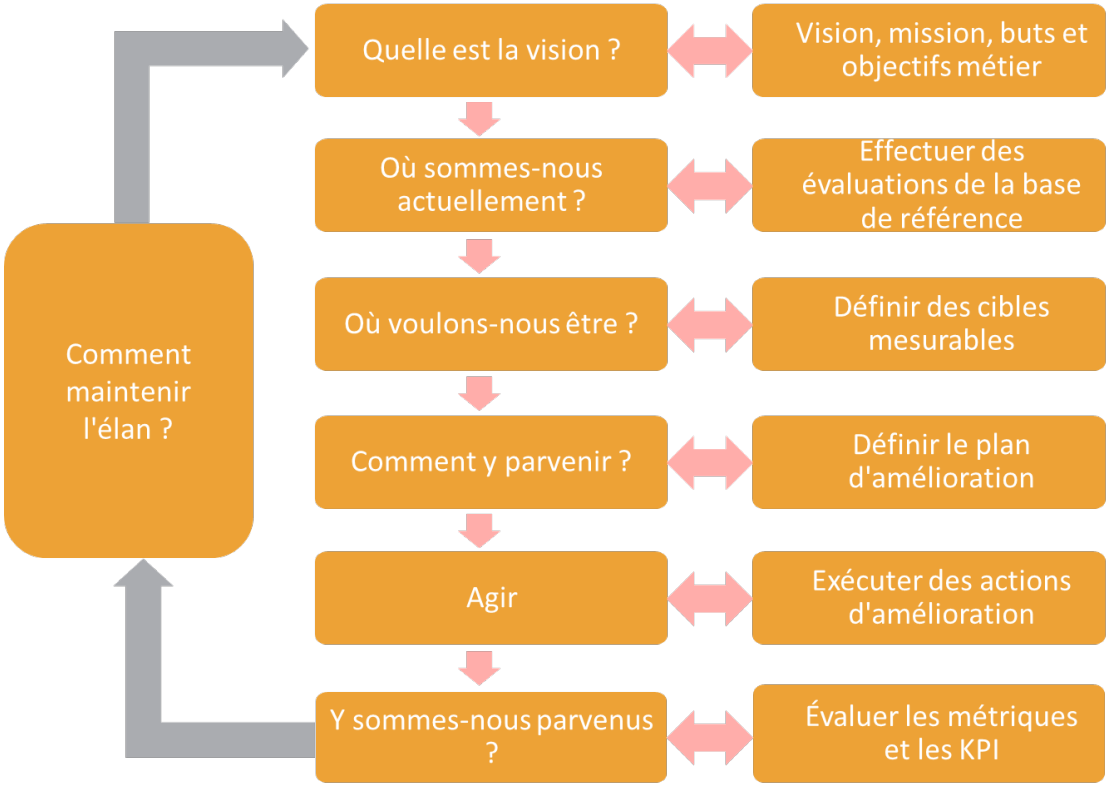
- **Le cycle DMAIC** fournit des conseils pour une amélioration continue à travers cinq étapes :

1. Définir : atteindre une compréhension du problème avec toutes les personnes impliquées
2. Mesurer : mesurer les données et rassembler des faits
3. Analyser : identifier les causes premières et définir des hypothèses
4. Améliorer : hiérarchiser et mettre en œuvre des actions
5. Contrôle : assurer la durabilité et la stabilité du processus amélioré

14. Amélioration continue

L'amélioration continue est essentielle en tant que solution aux problèmes de livraison de valeur commerciale et informatique. C'est un moyen de rester en phase avec la valeur demandée par les entreprises et avec le rythme avec lequel la concurrence émerge et le monde évolue.

Cela exige non seulement une amélioration constante, mais également une mesure stratégique de cette amélioration afin de concentrer votre attention sur les contraintes et les domaines sur lesquels elle aura le plus grand impact.



(Figure 4.3 The Continual Improvement Model – ITIL® Foundation, ITIL 4 edition, 2019)

14. Amélioration continue (suite)

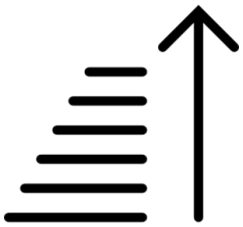
Registre d'amélioration continue (CIR)

Le **registre d'amélioration continue** est un document qui fournit structure et visibilité aux efforts d'amélioration continue, et qui enregistre et classe chaque possibilité d'amélioration en fonction du calendrier, du niveau d'effort, de l'urgence, de l'impact et de la priorité.

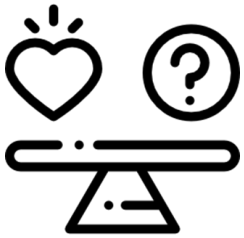
N°ID	Date d'ajout	Ajouté par	Niveau d'effort	Durée	Description	Raison	Urgence	Impact	Priorité

L'amélioration continue prend tout en charge dans DevOps

L'amélioration continue doit être mise en œuvre dans le but d'une amélioration prenant en charge le Full stack et les trois principes, avec des objectifs de :



Aligner les objectifs et les priorités autour de la création de valeur pour l'entreprise



Trouver un équilibre entre l'amélioration des délais et des coûts et l'amélioration de la qualité et de l'efficacité.

15. Antifragilité

L'antifragilité va au-delà de la résilience

15. Antifragilité (suite)

Faire de l'échec un outil d'apprentissage

En 2011, Netflix a lancé « Chaos Monkey », un logiciel qui simule une défaillance en arrêtant de manière aléatoire des serveurs de production.

THE SIMIAN ARMY

Cela a évolué en une collection d'outils de test sur le Cloud pour simuler d'autres défaillances potentielles que l'entreprise pourrait rencontrer en production. Cette collection d'outils a été nommée « L'armée simienne ».

En 2016, a été lancé Chaos Kong, une frappe de DDoS (Déni de Service Distribué) simulée qui a mis l'antifragilité de Netflix à l'épreuve ultime.

L'ingénierie du Chaos est une discipline qui vise avant tout à lutter contre la fragilité et à assurer des performances stables, même en cas d'urgence ou d'incidents ou de pannes imprévues.

> *Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capability to withstand turbulent conditions in production.*
>
> **The Principles Of Chaos Engineering**

Source : http://blog.cognitect.com/blog/2016/3/24/the-new-normal-embracing-failure-with-netflix-the-chaos-monkey-and-chaos-kong

Les 15 pratiques essentielles de DevOps