

Lexique du devops

Par Dimitri PIANETA

avril 2021

Sommaire

-A-	3
-B-	4
-C-	4
-F-	5
-G-	5
-I-	6
-K-	7
-L-	8
-M-	9
-O-	10
-P-	11
-R-	11
-S-	12
-T-	12
-V -	13
-W-	13

-A-

Agile (méthode) :

Une méthode agile est une méthode utilisée dans le cadre de la création et/ou du développement d'un produit qui est :

- itérative
- collaborative
- incrémentale

Le but d'une méthode agile est de livrer rapidement au client final un produit qui correspond à ses besoins et à ses attentes, de la meilleure qualité possible et pouvant évoluer facilement.

Agile manifeste (Agile Manifesto) : fonctionnement d'un système sans intervention humaine, automatisation d'une suite d'opération.

Le Manifeste Agile ou Agile Manifesto est une déclaration officielle déterminant les valeurs et les principes fondamentaux des méthodologies agiles. Publié en 2001, il a été rédigé par dix-sept experts américains du développement d'applications.

Ses quatre valeurs sont :

1. les individus et leurs interactions,
2. la fonctionnalité du produit final,
3. la collaboration avec le client,
4. l'adaptation au changement.

Le Manifeste Agile est défendu par l'association Agile Alliance.

Déclaration officielle des valeurs et des principes pour une approche itérative et axée sur l'individu dans le cadre d'un développement logiciel.

Antifragile :

Concept inventé par le professeur Nassim Taleb, il s'agit d'une propriété des systèmes qui se renforcent ou s'améliorent lorsqu'ils sont confrontés à une certaine dose de stress, de force, ou de choc.

Automatisation (automation) :

Fonctionnement d'un système informatique ou exécution d'un processus sans intervention humaine. L'automatisation permet de réduire grandement les risques d'erreurs, d'intégrer en temps réel de nouvelles fonctionnalités et de faire gagner du temps aux équipes de production.

Il s'agit de la technologie par laquelle un processus ou une procédure est exécuté sans intervention manuelle. Dans DevOps, l'automation permet de créer des rapports en temps réel, intégrant les outils utilisés par différentes parties prenantes, des workflows, intégrant des technologies pour rassembler les outils de différents domaines et dissocier les silos. -

-B-

Bottleneck (Lean) :

Une étape dans un processus qui limite la capacité totale du processus ou du système.

Build : concerne la conception de l'application.

Dans un plan *Build – Run – Change*, le *Build* est la phase de conception d'un système informatique. C'est à ce moment que l'on imagine l'application, ce qu'elle va permettre de faire, ses objectifs et la façon dont elle va être construite (combien de temps, quelles équipes, etc).

-C-

Change : concerne l'évolution de l'application.

Dans un plan *Build – Run – Change*, le *Change* constitue ce qui concerne l'évolution d'un système informatique. Ce sont les moments durant lesquels on fait évoluer l'application.

ChatOps :

L'utilisation de chats clients, de chatbots et d'outils de communication en temps réel pour faciliter la communication et l'exécution des tâches de développement logiciel et d'exploitation.

Conteneur (containers) : Un conteneur de développement logiciel regroupe les applications et leurs dépendances, ce qui permet de créer des applications dans lesquelles les environnements de test et réels peuvent être répliqués fidèlement.

Un conteneur est un environnement informatique dédié à une application qui lui fournit les ressources (fichiers, variables, bibliothèques) dont elle a besoin pour fonctionner.

Il permet de disposer d'un espace d'exécution isolé pour tester de manière saine des applications ou des logiciels en cours de développement et identifier facilement les potentiels risques et failles de sécurité.

Contrainte (Constraints) :

Une limitation ou restriction dans un système.

Contrainte (théorie) :

Une méthode permettant d'identifier le facteur limitant le plus important empêchant d'atteindre un objectif, puis d'améliorer systématiquement cette contrainte jusqu'à ce qu'elle ne soit plus le facteur limitant.

Culture : Tous les idées, valeurs, croyances, pratiques et comportements qui sont partagés par les employés d'une organisation. Chaque équipe DevOps met en place une politique de responsabilité partagée.

-D-

Definition of Done :

La *Definition of Done* est un l'accord de toutes les parties prenantes d'un projet sur un ensemble de spécifications et/ou de user stories qui doivent être achevées ou satisfaites pour qu'un système informatique soit considéré comme terminé.

Déploiement (Deployment) : La publication de mises à jour logicielles pour les utilisateurs. Dans les environnements DevOps, le déploiement est entièrement automatisé afin que les utilisateurs obtiennent les mises à jour dès qu'elles sont écrites et testées.

Le déploiement ou *deployment* est l'introduction coordonnée de mise à jour et de nouveaux services au sein d'un système informatique (application, logiciel ...). L'un des buts de l'approche DevOps est de rendre les déploiements automatiques pour réduire son temps et rendre les nouveautés accessibles en temps réel pour les utilisateurs de l'application informatique.

Dev (DevOps) :

Dans l'approche DevOps, *Dev* désigne les équipes de développeurs qui sont engagées dans le développement logiciel.

DevOps :

DevOps est une approche visant à rapprocher les équipes de développement logiciel (*Dev*) et celles d'administration des systèmes (*Ops*) dans le but d'automatiser toutes les étapes de la création d'un système informatique grâce à des méthodes de partage, d'automatisation, de test et de mesure.

L'approche DevOps repose notamment sur la livraison et l'intégration en continu permise grâce à la communication entre équipes *Dev* et équipes *Ops* **pour réduire le temps nécessaire à la production et au déploiement d'applications dans un double objectif de qualité et rentabilité.**

DevSecOps :

L'automatisation des tâches de sécurité essentielles en incorporant des contrôles et des processus de sécurité dans le flux de travail DevOps. L'objectif est d'intégrer le plus tôt possible la sécurité dans le processus afin de minimiser les vulnérabilités et les risques

-F-

Flow :

Façon dont les personnes et les productions évoluent et agissent au cours d'un processus.

Comment les personnes ou les produits traversent un processus. La priorité de DevOps concerne l'optimisation des systèmes en flux continu.

-G-

Gemba : Un mot japonais pour dire "le vrai lieu". En affaires, cela équivaut souvent à l'endroit où la valeur est créée.

Gestion de projet agile (Agile Project Management) :

Une méthode itérative et incrémentale de développement dans laquelle les développeurs travaillent de près avec les clients afin de délivrer très rapidement et très régulièrement des "produits" suffisamment aboutis pour être évalués et améliorés .

-I-

Idempotence : une opération est idempotente si elle a le même effet quelle soit appliquée une ou plusieurs fois. Les outils de gestion de configuration sont généralement idempotent.

En développement informatique, l'idempotence signifie qu'une opération produit le même résultat, qu'elle soit réalisée une ou plusieurs fois.

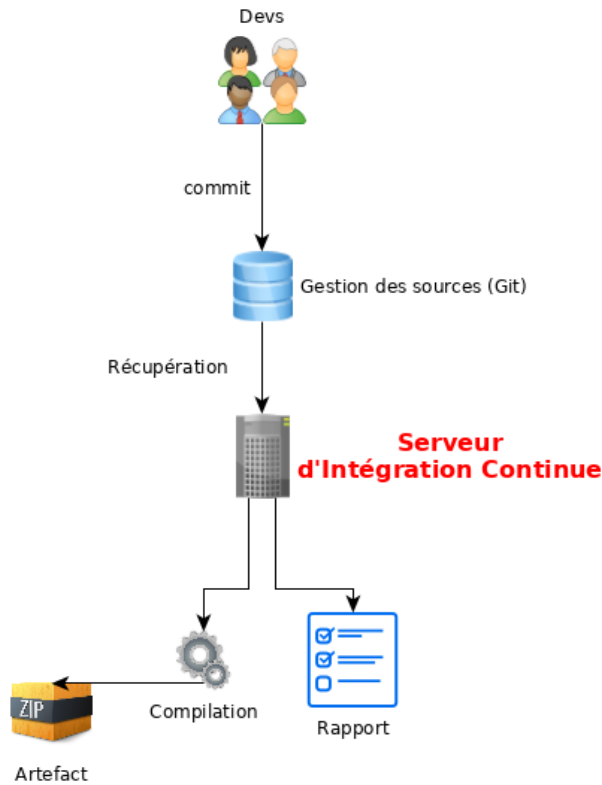
Integration continue (continuous integration CI) : chaque modification d'un code source entraîne une vérification de non-régression de l'application.

Une pratique de développement logiciel qui oblige les développeurs à fusionner leur code dans un système de contrôle de version partagé. L'objectif est de localiser et de corriger les bogues logiciels plus rapidement et de réduire le temps nécessaire à la publication des mises à jour logicielles.

L'intégration en continue consiste à **intégrer de manière continue les nouveautés et les changements apportés à un projet** de développement d'une application ou d'un système informatique et **de le tester en permanence** de manière automatique.

L'intégration continue permet donc de **repérer les failles dans le nouveau code** aussi rapidement que possible et de **réduire le temps total de développement informatique**.

Pour cela, une équipe de développement DevOps **intègrent leur code au projet dans une version partagée** au moins une fois par jour. Le code source est vérifié automatiquement à chaque fois qu'il est modifié.



Itérations (iterations) :

Une itération est un cycle de production court (analyse, programmation, test...) qui se conclut par une livraison interne ou externe et qui dure entre une semaine et un mois.

-K-

Kaizen :

Une philosophie d'entreprise japonaise axée sur l'amélioration continue des pratiques de travail, l'efficacité personnelle, etc. Le but est de trouver des moyens d'améliorer un flux de valeur complet conduisant à de meilleurs résultats pour les clients.

Kanban :

Méthode agile visant à réduire les délais et les coûts en se concentrant sur les demandes et la satisfaction du client pour réduire les coûts inutiles grâce à une amélioration continue.

Imaginée par Taiichi Ōno dans les années 1950 dans le cadre de la production d'automobiles au Japon, la méthode Kanban demande de se concentrer sur les demandes du marché. C'est donc **une méthode à flux tirés** à la différence des systèmes fondés sur les flux poussés.

La méthode Kanban repose sur une visualisation des flux de développement qui permet de suivre l'avancement, la planification et de prioriser les tâches.

Tableau Kanban (*Kanban Board*) :

Le *Kanban Board* est un outil qui permet de visualiser l'avancement des flux de travail et de prioriser les tâches à réaliser dans un planning. Les équipes agiles l'utilisent pour organiser et gérer leur travail.

Le tableau Kanban se divise généralement en trois colonnes :

- À faire
- En cours
- Réalisé

Selon leur avancement, les membres d'une équipe agile font glisser les tâches et missions représentées dans le tableau sous forme d'**étiquettes** ("Kanban", en japonais).

Kanban Board :

Un outil Kanban qui aide les équipes à organiser, visualiser et gérer leur travail.

Kata :

En japonais, c'est le conditionnement culturel ou l'idée de faire les choses «correctement». C'est un moyen systématique d'atteindre des objectifs et de relever des défis qui peuvent être utilisés dans une organisation.

-L-

Lean :

Approche de la production se concentrant sur l'élimination du gaspillage. Issue de l'industrie automobile japonaise dans laquelle elle a été intégrée dans les années 1970, le Lean management vise à maximiser la qualité des produits et la performance des processus en utilisant un minimum de ressources (humaines, temporelles et économiques).

L'approche Lean recherche une performance maximale que l'on tente d'atteindre par l'amélioration des procédures et l'élimination du superflux.

Lean IT :

Il s'agit d'appliquer les idées clés du Lean au développement et aux opérations liées aux produits et services informatiques.

Livraison continue (*continuous delivery*) : Une méthodologie visant à garantir que les logiciels sont toujours prêts pour la publication tout au long de leur cycle de vie. Le processus de déploiement devient itératif, fournissant des versions plus fréquentes à l'utilisateur final.

Approche incrémentale de la production d'applications et de logiciels visant à réduire le temps de mise à disposition et les coûts de production.

La livraison continue, ou *continuous delivery*, est une pratique clé de l'approche DevOps qui consiste à intégrer au fur et à mesure les nouveautés et les modifications apportées au code source du système informatique. Elle est rendue possible si elle est réalisée par **un processus de développement, de test et de déploiement automatique : le Continuous Delivery Pipeline.**

Une fois que le développeur a intégré son code au répertoire source, il est testé et déployé automatiquement, rendant les mises à jours disponibles immédiatement.

-M-

Microservices :

Service d'une application informatique indépendant des autres. Dans une architecture orientée service, une application est construite comme un ensemble de microservices qui fonctionnent de manière autonome. Ainsi, si l'un d'eux est en panne, il ne met pas en péril la totalité de l'application.

Modèle CALMS :

L'essence de DevOps : Culture, Automatisation, Lean, Mesure & Partage. Cette approche peut être utilisée pour évaluer si une organisation est prête à s'engager dans un processus DevOps.

- *Culture* : Comment défendre le ROI (retour sur investissement) prévu lié à l'automatisation d'un processus au sein de l'équipe DevOps ? Pour qu'un projet aboutisse, il doit bénéficier de l'appui des dirigeants de l'entreprise. C'est la partie cruciale de la 'culture', afin de ne pas investir dans la technologie pour la technologie, mais pour répondre à un besoin commercial.
- *Automatisation* : L'automatisation est au centre du DevOps. Cependant, à tort, on limite le DevOps à cet aspect. Pourtant, l'automatisation des processus ne représente que 20 à 25% du DevOps.

Sont automatisées les tâches répétitives : exécution des tests (tests fonctionnels et non fonctionnels), déploiement, provisionning / deprovisionning d'infrastructure, configuration, etc.

L'automatisation permet d'éviter les erreurs (notamment humaines) et les failles, d'améliorer la productivité et les performances et de fluidifier la livraison du produit au client.

- *Lean* : Dans le cadre de CALMS, la composante Lean est relativement simple à évaluer, puisqu'elle repose sur la documentation des économies de coûts et l'élimination des ressources.

La maximisation de la valeur client et la minimisation des gaspillages sont ici adaptés à l'IT et au DevOps plus spécifiquement.

La pensée Lean invite les équipes DevOps à identifier les tâches qui créent de la valeur durant le cycle et optimiser les autres tâches. En développement par exemple, les tâches créant de la valeur sont le codage d'une fonctionnalité et la livraison de cette fonctionnalité au client. Les autres tâches (test, test pré-prod, contrôle du code, etc...), quoique très importantes, n'apportent pas de valeur directe au client. Elles devraient être optimisées et être améliorées (ex : réduction du temps associé à ces tâches).

Une variante du Lean est le Lean Software Development qui a 7 principes fondateurs

- Eliminer les gaspillages (retard de livraison, fonctionnalités non utilisées, les bugs, etc.)
- Favoriser l'apprentissage (expérimentation, créativité)
- Décider le plus tard possible (au temps opportun en tenant compte de l'expérience et des informations collectées)
- Livrer rapidement
- Responsabiliser l'équipe (pour qu'elle soit engagée et obtienne de meilleures performances)
- Construire un produit de qualité
- Optimiser le système dans son ensemble

- *Mesure* : S'il est relativement facile de mesurer le succès d'un déploiement logiciel une fois celui-ci terminé, c'est beaucoup plus difficile d'essayer de le mesurer dans le cadre d'un modèle de précision pour le futur. Les mesures devraient permettre de localiser rapidement les erreurs au cours du post-déploiement afin de corriger le code si nécessaire également très rapidement.

Sans la capacité à obtenir des informations sur l'atteinte des objectifs, il est difficile d'évaluer les performances et de mettre en place les plans d'action pour s'améliorer.

Les indicateurs de performances à suivre dans le cadre du DevOps sont de différentes natures et permettent d'avoir une vue d'ensemble sur les performances de la démarche. On évoque notamment la satisfaction client, la productivité, la qualité ou encore la satisfaction collaborateur.

- *Sharing*(partage) : Si DevOps signe la fin du développeur solitaire, le piège est de continuer son propre travail sans penser suffisamment à l'équipe. C'est pourquoi le partage doit chevaucher la culture afin que les équipes travaillent ensemble et que la communication soit une réalité.

Muda :

Un mot japonais signifiant "futilité; inutilité; gaspillage." Muda est l'un des trois types de gaspillage dans la philosophie Lean.

Mura :

Un mot japonais signifiant "inégalité, irrégularité, manque d'uniformité". C'est l'un des trois types de gaspillage selon la façon de penser les processus Lean.

Muri :

Un mot japonais signifiant "déraisonnable; impossible; au-delà de son pouvoir ou trop difficile". C'est l'un des trois types de gaspillage selon la façon de penser les processus Lean.

-O-

Ops (DevOps) :

Dans l'approche DevOps, *Ops* désigne les équipes qui sont engagées dans l'administration des systèmes et des infrastructures.

Orchestration : processus automatique de gestion d'un système informatique.

En informatique, l'orchestration désigne les processus automatiques de gestion et d'organisation des systèmes informatiques.

Dans l'approche DevOps, l'orchestration est la façon dont se coordonnent de manière automatique les outils employés.

L'orchestration des conteneurs désigne la façon dont les conteneurs d'applications d'un projet de développement s'organisent et se coordonnent automatiquement.

Outils devops :

- Outils de gestion de configuration
 - Puppet
 - Ansible
 - Saltstack
 - Chef
- Intégration continue
 - Jenkins
 - Gitlab-ci
- Orchestration des tâches
 - Rundeck
 - Ansible Tower
- Infrastructure As Code
 - Terraform
- Docs as Code
 - AsciiDoctor
 - Markdown
 - ReSTructured Text

Organisation agile :

Une entreprise flexible capable de répondre et de s'adapter aussi bien aux opportunités qu'aux menaces, prévues ou imprévues.

-P-

Provisioning : processus d'allocation automatique de s ressources.

L'approvisionnement, ou provisionnement, ou *provisioning*, désigne **les processus d'allocation automatique de ressources** aux serveurs, aux applications ou aux utilisateurs, par exemple.

-R-

Récit utilisateur (user story) :

Une *user story* est un besoin exprimé par le client ou l'utilisateur d'un système informatique.

Il permet de concevoir les fonctionnalités nécessaires pour y répondre, à condition d'être formulé de manière suffisamment précise. Pour ce faire, il est généralement exprimé sous la forme :

*En tant que **QUI**, je veux **QUOI** dans le but de **POURQUOI**.*

Release :

Une release est la mise à disposition d'une nouvelle version d'un logiciel ou d'une application informatique agrémentée d'une ou plusieurs modifications. Les nouveautés peuvent par exemple être des améliorations apportées en terme de sécurité, de stabilité ou de fonctionnalités.

Run : concerne la maintenance de l'application.

Dans un plan *Build – Run – Change*, le *Run* constitue ce qui concerne l'exécution et la maintenance d'un système informatique.

-S-

Scrum :

Méthodologie agile incrémentale basée sur la collaboration avec le client ayant pour objectif de concentrer les efforts d'une ou plusieurs équipes sur les priorités pour le client. Le but de cette approche de la gestion de projet agile est de délivrer le produit de la meilleure qualité.

Dans une équipe utilisant la méthode Scrum, le travail est divisé en sprints. Au sein de l'organisation, le Scrum Product Owner représente les intérêts du client.

Serverless :

Un modèle de cloud computing dans lequel le fournisseur de cloud gère de manière dynamique l'allocation des ressources de la machine. La tarification est basée sur la quantité réelle de ressources consommées par une application. Ceci est souvent appelé "Function as a service".

Sprint :

Cycle de travail d'une durée courte (généralement comprise entre deux et quatre semaines) durant lequel une ou plusieurs fonctionnalités sont élaborées, développées testées et déployées.

C'est un cycle qui fait partie d'un processus itératif et incrémental, un modèle agile.

Un sprint s'achève par une livraison et dans certains cas, une rétrospective.

-T-

Test Driven Development (TDD) :

Pratique de développement logiciel basé sur des cycles de production courts et dont l'amélioration de la qualité est basée sur les résultats de tests automatiques définis en amont.

Time to Value :

Le Time to Value désigne le temps nécessaire à une organisation pour permettre à une fonctionnalité d'avoir de la valeur. Certaines fonctionnalités ont une valeur immédiate alors que d'autres sont développées dans le but d'acquérir de la valeur dans le futur.

The Three Ways :

Un ensemble de principes proposé par Gene Kim, Directeur technique, auteur et chercheur primé, pour définir ce qu'est vraiment DevOps.

- accélérer le flux de travail, depuis les activités commerciales (à travers le développement) jusqu'aux opérations et au client.
- augmenter le nombre et la rapidité des boucles de rétroaction dans un flux

- développer et promouvoir une culture où l'expérimentation continue et l'apprentissage sont encouragés.

Toolchain :

L'usage d'un set intégré d'outils spécifiques à une tâche pour automatiser un processus du début à la fin. Par exemple le test de code automatisé, la décharge et le déploiement.

-V -

Value Stream Mapping :

Un outil de Lean qui visualise le flux de données, de matériaux et de travail dans un processus en mettant l'accent sur l'identification et la quantification des déchets.

Vélocité (agile) :

Une mesure de la quantité de travail effectuée dans un intervalle prédéfini.

-W-

Waste (Lean) :

Tout ce qui n'ajoute pas de valeur à un produit. (Cf. Muda, Muri, Mura.)

Waterfall (Software Development) :

Une approche linéaire et séquentielle de la conception et du développement de logiciels, dans laquelle les progrès sont perçus comme un flux continu et séquentiel.

Un modèle en cascade ou waterfall, aussi désigné "cycle en V", est une méthode de gestion de projet linéaire dans lequel se succèdent différentes phases d'analyse, d'élaboration, de production et de tests avant la livraison finale.

Dans ce modèle, les phases de travail dépendent de celles d'avant.

Work in Progress (WIP) :

Tous les travaux qui ont été commencés mais qui ne sont pas encore terminés.