



Comprendre les métriques DORA

Votre guide complet



Sommaire

Introduction	3
À propos de l'auteur	4
D'où viennent les métriques DORA?	5
Quelles sont les métriques DORA?	6
Fréquence de déploiement	7
Délai nécessaire aux changements	10
Taux d'échec des changements	13
Temps de récupération d'un échec de déploiement	16
Les avantages de suivre les métriques DORA	19
Les plus grands défis des métriques DORA	21
Ce qu'il faut retenir	22
Par où commencer?	22
Comment implanter les métriques DORA?	23
Ressources supplémentaires	23
À propos d'Axify	24

Introduction

Métriques DORA 101

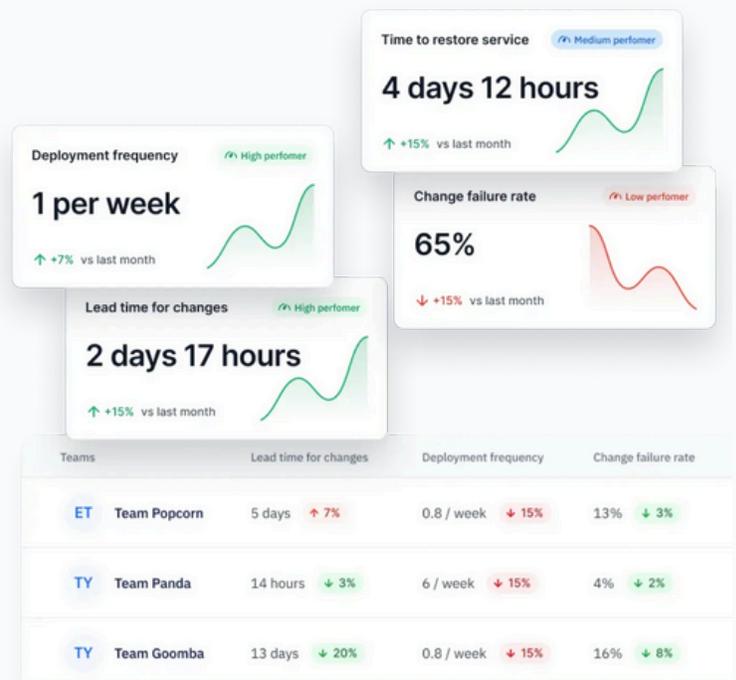
Dans un contexte où le développement de logiciels et la rapidité de livraison sont essentiels à la réussite des entreprises, il n'a jamais été aussi crucial de comprendre et d'optimiser les indicateurs clés qui déterminent les performances. Le rapport State of DevOps, publié chaque année par l'équipe DORA (DevOps Research and Assessment), est une ressource incontournable pour l'identification de ces indicateurs de performance essentiels. Ces métriques, communément appelées métriques DORA, fournissent un cadre clair et opérationnel aux organisations qui s'efforcent de mettre en place des processus de livraison logicielle performants.

Ce guide se penche sur les quatre indicateurs de base de DORA : fréquence de déploiement, délai nécessaire aux changements, temps de récupération d'un échec de déploiement et taux d'échec des changements. En expliquant chaque mesure en profondeur, nous souhaitons donner à vos équipes les moyens de mesurer, d'analyser et

d'améliorer leurs performances de livraison logicielle.

Grâce à la théorie et aux applications concrètes, vous comprendrez comment exploiter les métriques DORA pour améliorer l'efficacité, la qualité et la fiabilité de vos processus de développement.

Commençons cette aventure pour élever vos pratiques de développement logiciel au prochain niveau!



À propos de l'auteur

Voici Alexandre

Au fil des ans, Alexandre a bâti la culture d'ingénierie moderne qui fait la renommée de Nexapp (l'équipe derrière Axify!). Il s'est également impliqué dans l'industrie du développement logiciel en partageant ses connaissances avec la communauté technologique de Québec et la prochaine génération de développeurs. Son objectif est d'aider le plus grand nombre d'équipes de développement possible à atteindre une performance de livraison élite et à se concentrer sur l'utilisateur grâce à l'expérimentation rapide de produits.

Depuis 2019, il a offert des présentations à plusieurs reprises aux Agile Tour de Québec et de Montréal et a été conférencier au DevOps Day de Montréal en 2024. Il est toujours partant pour parler de l'un de ses sujets préférés, les métriques DORA, au plus grand nombre de personnes possible.



Chef de produit et directeur
de la technologie, Axify

D'où viennent les métriques DORA?

DORA est un acronyme pour DevOps Research and Assessment, une équipe qui étudie activement ce qui différencie une équipe de développement très performante d'une équipe peu performante. Dans le cadre d'un programme de sept ans, acquis par Google en 2018, ce groupe de recherche a analysé les pratiques et les capacités DevOps de plus de 32 000 professionnels du domaine et a pu identifier en 2020 quatre indicateurs clés pour mesurer les performances en matière de développement et de livraison de logiciels.

Outre un rapport DevOps officiel publié annuellement, l'équipe a également publié un livre blanc sur le retour sur investissement de la transformation DevOps, ainsi que le livre «*Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*», coécrit

par Dre Nicole Forsgren, cofondatrice de l'équipe DORA et actuellement partenaire à Microsoft Research.

Puisqu'il existe de nombreux *frameworks* et méthodologies visant à améliorer la façon dont les équipes de développement bâtissent des produits et services logiciels, DORA souhaite mettre en lumière ce qui fonctionne et ce qui ne fonctionne pas de manière scientifique.



Depuis leur publication, les métriques DORA sont utilisées par les équipes DevOps partout dans le monde pour mesurer leur performance et savoir comment elles se comparent aux équipes de haute performance.

Quelles sont les quatre métriques DORA?



Fréquence de déploiement

Fréquence à laquelle une équipe met un item en production.



Délai nécessaire aux changements

Temps requis pour qu'un *commit* passe en production.



Taux d'échec des changements

Pourcentage de déploiements entraînant un échec en production.



Temps de récupération d'un échec de déploiement

Temps nécessaire à une équipe pour se remettre d'une défaillance en production.

À haut niveau, la fréquence de déploiement et le délai nécessaire aux changements mesurent la **vitesse**, tandis que le taux d'échec des changements et le temps de récupération d'un échec de déploiement mesurent la **stabilité**. En mesurant ces indicateurs, et en itérant continuellement pour les améliorer, une équipe peut obtenir de meilleurs résultats commerciaux. En 2021, l'équipe DORA a ajouté une cinquième métrique, la fiabilité, à la liste des éléments ayant un impact sur les performances organisationnelles. Dans cet article, nous couvrirons les quatre premières métriques, soit la fréquence de déploiement, le délai nécessaire aux changements, le taux d'échec des changements et le temps de récupération d'un échec de déploiement.

Les études de l'équipe DORA démontrent que les équipes aux performances de livraison élevées sont deux fois plus susceptibles d'atteindre ou de dépasser leurs objectifs de performance organisationnelle.

Fréquence de déploiement

Comme son nom l'indique, la fréquence de déploiement fait référence à la fréquence des mises en production : elle mesure la fréquence à laquelle une équipe déploie un changement en production.



Les équipes de développement qui ont de meilleures performances de livraison ont tendance à effectuer des livraisons plus petites et beaucoup plus fréquentes, afin de fournir de la valeur plus fréquemment aux utilisateurs, améliorer la fidélisation des clients et garder une longueur d'avance sur la concurrence.

Indicateur de référence

À quelle fréquence votre organisation déploie-t-elle du code en production ou le rend-elle accessible aux utilisateurs finaux?

Performance élite	Haute performance	Performance moyenne	Performance faible
Sur demande	Entre une fois par jour et une fois par semaine	Entre une fois par semaine et une fois par mois	Entre une fois par semaine et une fois par mois

Source : [2023 Accelerate State of DevOps](#), Google.

Pourquoi mesurer la fréquence de déploiement?

La fréquence de déploiement, comme son nom l'indique, calcule le nombre de déploiements effectués par l'équipe pour une période donnée (par mois, semaine ou jour, par exemple). Une fréquence de déploiement élevée est un bon indicateur de la capacité de l'équipe à apporter des changements.

De plus, une fréquence de déploiement stable (donc une cadence de déploiement régulière) peut être le reflet d'une équipe agile livrant de la valeur en continu, ce qui permet de réduire la boucle de rétroaction, un avantage inestimable pour l'équipe et les utilisateurs.

Causes potentielles d'une fréquence de déploiement faible

- ✓ Projets trop complexes, gros changements ou développement de fonctionnalités au complet avant de déployer en production ;
- ✓ Pipeline de l'intégration continue (*Continuous Integration* ou CI) incomplet ou pas assez automatisé ;
- ✓ Incapacité à envoyer en production sur demande (en un clic) ;
- ✓ Goulots d'étranglement, dépendances dans le processus ou avec une autre équipe (comme une équipe DevOps dédiée au déploiement en production, par exemple) ;
- ✓ Nécessité d'un contrôle de qualité ou de déploiement manuels, ou manque d'automatisation ;
- ✓ Outils inefficaces, migration des bases de données non automatisées, mauvaise gestion de la configuration, environnement de production différent des autres environnements ou besoin de manipulations manuelles ;
- ✓ Déploiement de plusieurs items en lot à la fin d'un cycle.

Comment améliorer la fréquence de déploiement?

Peu importe où se situe votre équipe dans le tableau ci-haut, voici quelques pistes pour vous améliorer :

- ✓ Automatisation complète des processus de déploiement ;
- ✓ Mise en place de plus de tests automatisés sur les comportements du système, dans le but d'avoir une plus grande confiance envers les changements (*shift left testing*) et être en mesure de les envoyer directement en production ;
- ✓ Décortiquer le travail à effectuer en plus petits items, tant pour les tâches que les pull requests ;
- ✓ Réduire le nombre d'environnements intermédiaires.

Fréquence de déploiement d'une équipe de développement sur une période de deux mois dans Axify. Avec 46,7 déploiements par semaine, l'équipe déploie les changements en production plus de neuf fois par jour.



Délai nécessaire aux changements

Afin de mesurer le délai nécessaire aux changements, deux données sont nécessaires : l'heure exacte du premier *commit* et l'heure exacte du déploiement dans lequel il a été fait. La durée moyenne est ensuite utilisée comme indicateur de performance, les équipes qui ont une meilleure performance de livraison sont capables de passer du *commit* à la livraison en production en moins d'un jour.



Cet indicateur mesure le temps écoulé entre le premier *commit* et son déploiement en production. Il est utilisé pour mieux comprendre le temps de cycle de l'équipe et analyser sa réactivité face à l'évolution constante des demandes et des besoins des utilisateurs. Plus le délai nécessaire aux changements est bas, plus l'équipe est réactive et ajoute de la valeur rapidement.

Indicateur de référence

Combien de temps faut-il pour passer du premier commit au code livré en production?

Performance élite	Haute performance	Performance moyenne	Performance faible
Moins d'un jour	Entre un jour et une semaine	Entre une semaine et un mois	Entre une semaine et un mois

Pourquoi mesurer le délai nécessaire aux changements?

Le délai nécessaire aux changements est un indicateur de la rapidité avec laquelle une équipe livre des changements aux utilisateurs. Il représente l'efficacité du *flow*, la complexité du code et la capacité de l'équipe. Des délais de livraison plus courts sont préférables, car ils permettent une boucle de rétroaction plus courte sur ce qui est développé et de corriger le tir plus rapidement. Les déploiements peuvent être retardés

pour diverses raisons, notamment la peur de l'échec au déploiement ou le refus d'une livraison partielle : il est donc important que les équipes de développement aient une idée précise du temps nécessaire pour mettre les changements en production. Cette mesure est particulièrement utile pour les entreprises ou gestionnaires devant suivre plusieurs contextes ou équipes, puisqu'elle devient un indicateur de référence.

Causes potentielles d'un délai nécessaire aux changements élevé

- ✓ Goulots d'étranglement et dépendances dans le processus ;
- ✓ Beaucoup de temps d'attente où les tâches ne sont pas activement travaillées ;
- ✓ Manque d'automatisation de déploiement ou difficile d'aller en production rapidement ;
- ✓ Revues de code trop lourdes et longues à effectuer ;
- ✓ Manque de tests automatisés sur les comportements (compensé par des tests manuels faits par des humains) ;
- ✓ *Legacy code* ou aucun temps alloué à rembourser la dette technique ;
- ✓ Dérapage du champ d'application (*scope creep*) et suringénierie.



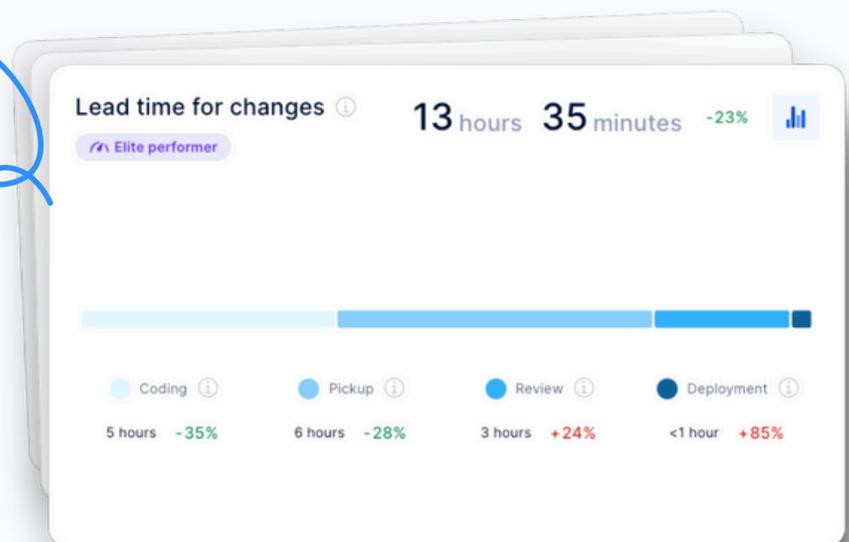
Analyser les étapes de votre processus pour cibler les opportunités d'automatisation peut être bénéfique pour votre équipe. En réduisant les manipulations humaines (tant au niveau des tests que des automatisations), vous contribuez à accélérer la cadence de livraison. Ajoutez à cela un remboursement progressif de la dette technique et un meilleur découpage des items, et vous serez sur la bonne voie pour réduire le délai nécessaire aux changements!

Comment améliorer le délai nécessaire aux changements?

Peu importe où se situe votre équipe dans le tableau ci-haut, voici quelques pistes pour vous améliorer :

- ✓ Encourager la collaboration et la réduction du travail en cours (WIP) et du nombre de revues effectuées en parallèle ;
- ✓ Mettre en place de processus de déploiement et de révision plus automatisés ;
- ✓ Diviser les tâches et fonctionnalités en petits items plus faciles à gérer ;
- ✓ Analyser le temps passé par étape du pipeline de développement afin d'identifier les goulots d'étranglement ;
- ✓ Éviter de dépendre d'une personne externe à l'équipe pour les révisions de code ;
- ✓ Favoriser le *trunk-based development* pour apporter de plus petits changements fréquemment ;
- ✓ Encourager le *Test-Driven Development* (TDD) ou investir davantage dans des tests automatisés qui sont orientés sur les comportements.

Le délai nécessaire aux changements d'une équipe de développement dans Axify. Certaines phases s'allongent tandis que d'autres se raccourcissent, mais le processus global est 23 % plus rapide.



Taux d'échec des changements



Cette métrique capte le pourcentage de modifications apportées à un code qui ont ensuite entraîné des incidents, des retours en arrière ou tout autre type d'échec de production. Selon le rapport DORA, les entreprises les plus performantes se situent autour de 5%.

La partie la plus délicate pour la plupart des équipes consiste à définir ce qu'est un «incident». Il peut s'agir d'une non-disponibilité du système (partielle ou complète), d'une fonctionnalité non utilisable, d'un bogue critique, majeur ou tout type de bogue. Selon DORA, il s'agit de changements apportés à la production ou aux versions destinées aux utilisateurs qui entraînent une dégradation du service (par exemple, une détérioration ou une interruption du service) et qui nécessitent ensuite des mesures correctives (par exemple, un *hotfix*, un *rollback*, un *fix forward* ou un *patch*). Une définition trop large ou trop restrictive pourrait nuire à l'équipe et la réponse dépend fortement du contexte de celle-ci.

Indicateur de référence

Quel pourcentage des modifications apportées à la production ou aux utilisateurs finaux entraîne une dégradation du service?

Performance élite	Haute performance	Performance moyenne	Performance faible
5%	10%	15%	64%

Pourquoi mesurer le taux d'échec des changements?

Il s'agit d'un indicateur de qualité et de stabilité du code. Cette métrique montre le pourcentage de changements effectués qui entraînent un incident. Comme mentionné précédemment, le type d'incident considéré pour cette métrique est propre à chaque équipe. Mais il est important de comprendre qu'un taux d'échec élevé implique une augmentation du

temps passé à retravailler sur des fonctionnalités existantes, donc une diminution du temps alloué à livrer de la nouvelle valeur aux utilisateurs. Bien que ce ne soit pas une métrique de temps à proprement parler, le taux d'échec des changements peut avoir des implications importantes sur votre capacité à créer de la valeur rapidement.

Un faible taux d'échec des changements montre que l'équipe identifie les erreurs et les bogues d'infrastructure avant le déploiement du code. C'est un signe que le processus de déploiement est solide et que la livraison de logiciel est de haute qualité. L'objectif de l'équipe DevOps devrait être de réduire le taux d'échec des modifications afin de garantir la disponibilité et le bon fonctionnement du logiciel.

Causes potentielles d'un taux d'échec des changements élevé

- ✓ Absence de tests automatisés, trop de tests manuels, tests inefficaces ou absents avant le déploiement ;
- ✓ Problèmes dans le processus de développement ;
- ✓ Gros changements qui demandent de longues révisions et où il y a peu de commentaires ;
- ✓ Code de mauvaise qualité ou couplage trop élevé ;
- ✓ Difficulté à maintenir et introduire du nouveau code ;
- ✓ Changements d'infrastructure non reproductibles.

Comment améliorer le taux d'échec des changements?

Peu importe où se situe votre équipe dans le tableau ci-haut, voici quelques pistes pour vous améliorer :

- ✓ Réduire le travail en cours (WIP) dans les itérations ;
- ✓ Favoriser les activités sociales de développement (ex. : *pair programming*, revue de code synchrone) ;
- ✓ Augmenter la confiance envers nos tests automatisés, en écrire davantage et encourager le TDD ;
- ✓ Travailler avec des itérations plus petites ;
- ✓ Utiliser des outils d'analyse de code statique dans votre intégration continue ;
- ✓ Créer une routine de surveillance d'équipe afin d'augmenter l'imputabilité de l'équipe face à la disponibilité et la fiabilité du système ;
- ✓ Créer une boucle de rétroaction récurrente ;
- ✓ Automatiser les migrations de base de données ou de création d'environnement.



Les équipes qui ne déploient pas beaucoup de changements verront moins d'incidents, mais cela ne signifie pas nécessairement qu'elles ont plus de succès avec les changements qu'elles déploient. Les équipes qui déploient fréquemment peuvent voir un nombre plus élevé d'incidents, mais si le taux d'échec des changements est faible, ces équipes auront un avantage en raison de la rapidité de leurs déploiements et de leur taux de réussite global.



Temps de récupération d'un échec de déploiement

Le temps de récupération d'un échec de déploiement mesure le temps nécessaire à un service pour rebondir après un incident ou une dégradation du service. Quel que soit le niveau de performance d'une équipe, des pannes ou des incidents imprévus se produiront. Et comme les incidents ne peuvent être évités, c'est vraiment le temps qu'il faut pour restaurer ou récupérer un système ou une application qui fait la différence.



Cette métrique est importante, car elle encourage les développeurs à construire des systèmes plus fiables, disponibles et résilients. Elle est généralement calculée en suivant le temps moyen entre le signalement de l'incident et le moment où sa restauration est déployée. Pour les équipes les plus performantes, il s'agit d'être capable de restaurer le service en moins d'une heure.

Indicateur de référence

Combien de temps faut-il pour restaurer le service après une panne, une dégradation ou l'introduction d'un bogue ayant un impact sur les utilisateurs?

Performance élite	Haute performance	Performance moyenne	Performance faible
Moins d'une heure	Moins d'un jour	Entre un jour et une semaine	Entre un et six mois

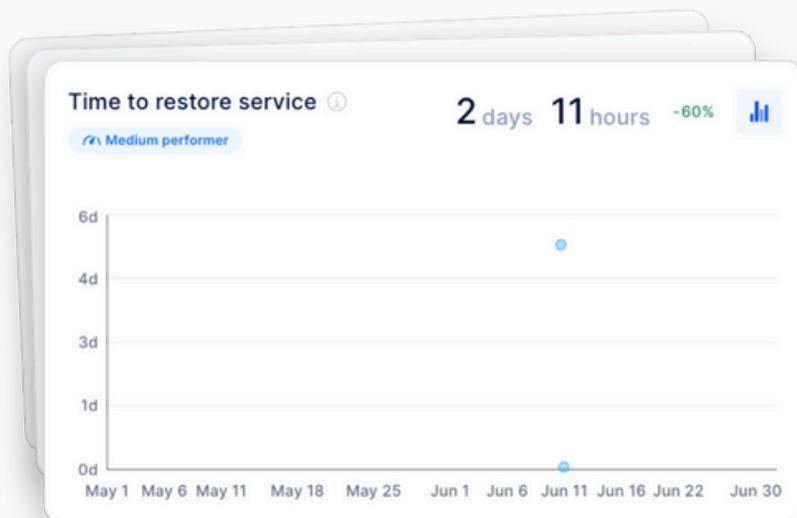
Pourquoi mesurer le temps de récupération d'un échec de déploiement?

Le temps de récupération d'un échec de déploiement indique le temps de réponse d'une équipe et l'efficacité du processus de développement. Lorsqu'il est bas, il démontre qu'une équipe réagit et résout les problèmes rapidement pour garantir la disponibilité et le bon fonctionnement du produit.

De plus, la capacité d'une équipe à se rétablir rapidement peut avoir un impact positif sur le niveau de confiance des utilisateurs et amener plus d'aisance de la part des dirigeants face aux expérimentations. C'est donc un avantage concurrentiel non négligeable!

Causes potentielles d'un temps de récupération d'un échec de déploiement élevé

- ✓ Absence de tests automatisés ;
- ✓ Dépendance vers un membre externe à l'équipe ou manque d'accès à certaines composantes du système ;
- ✓ Aucun changement de focus dans l'équipe pour résoudre l'incident ;
- ✓ Processus de gestion des incidents inefficace ;
- ✓ Difficulté de déployer en production rapidement ;
- ✓ Aucun processus de restauration à une version antérieure (*rollback*) ;
- ✓ Outils inefficaces.



Comment améliorer le temps de récupération d'un échec de déploiement?

Peu importe où se situe votre équipe dans le tableau ci-haut, voici quelques pistes pour vous améliorer :

- ✓ Favoriser une surveillance en continu et améliorer l'observabilité ;
- ✓ Donner la priorité à la restauration lorsqu'une défaillance se produit ;
- ✓ Essayer les *feature flags* pour être en mesure de désactiver rapidement un changement sans causer trop de perturbations ;
- ✓ Livrer de petites itérations pour qu'il soit plus facile de découvrir et de résoudre les problèmes ;
- ✓ Investir davantage dans les tests automatisés et utiliser une stratégie de tests orientés sur les comportements ;
- ✓ Documenter le processus de gestion des incidents et former les membres de l'équipe sur la façon de réagir en cas d'incidents ;
- ✓ Lorsque c'est applicable, automatiser plusieurs étapes du processus de gestion des incidents et désigner des responsables pour les étapes qui doivent être réalisées manuellement.



Équilibrez vitesse et stabilité! Permettez-vous parfois de ralentir pour mieux accélérer, en réduisant par exemple le travail en cours (WIP). Évitez de mettre en production des changements soudains au détriment d'une solution de qualité. Plutôt que de déployer une solution rapide, assurez-vous que le changement que vous apportez est durable et testé. Vous devriez suivre le temps moyen de récupération d'un échec de déploiement dans le temps pour voir comment votre équipe s'améliore et viser une croissance régulière et stable.

Les avantages de suivre la progression des métriques DORA



Amélioration continue

Pour s'améliorer, il faut deux éléments : un point de départ et un objectif à atteindre. Les métriques DORA donnent aux équipes de développement des objectifs concrets, qui peuvent ensuite être décomposés en résultats clés. En sachant quelles données suivre au fil du temps, les développeurs vont ajuster leur comportement pour améliorer la mesure de leur côté, ce qui aura un impact sur l'efficacité de l'équipe et des processus.

De plus, les variations dans les résultats des métriques DORA vous aideront à cibler rapidement les points à améliorer en ce qui concerne le processus de livraison et à repérer les problèmes tels que l'inefficacité des tests.



Meilleure visibilité

Les métriques DORA sont des métriques fiables résultant d'études scien-

tifiques à jour. Comme les quatre métriques sont reliées entre elles, il est difficile de les tricher : vous avez donc un portrait réaliste de votre situation. Elles permettent aussi de comparer votre équipe au reste de l'industrie, d'identifier les possibilités d'amélioration et d'apporter des changements pour optimiser vos résultats à plusieurs niveaux.



Aide à la prise de décision

Les entreprises qui rationalisent leur processus de développement et de livraison augmentent la valeur des logiciels qu'elles développent et sont plus performantes à long terme. Le suivi des performances à l'aide des métriques DORA permet aux équipes DevOps de prendre des décisions fondées sur des données, et non sur des intuitions.

De plus, les métriques DORA aident à aligner les objectifs de développement sur les objectifs commerciaux. Du point de vue de la gestion des produits, elles permettent de savoir comment et quand les équipes de

développement peuvent répondre aux besoins des clients.



Production de valeur

Ces dernières années, la gestion de la chaîne de valeur (Value Stream Management) est devenue un élément important du développement logiciel. Dans ce contexte, les métriques DORA jouent un rôle indispensable, car les équipes qui se mesurent ont tendance à s'améliorer en continu. Ainsi, lorsque les équipes DevOps utilisent les métriques DORA, elles constatent généralement une augmentation de la valeur au fil du temps.

En tant qu'ensemble de repères DevOps éprouvés qui sont devenus la norme dans le secteur, les métriques DORA fournissent une base pour ce processus. Elles identifient les points d'inefficacité ou de gaspillage, et vous pouvez utiliser ces informations pour rationaliser et réduire les goulots d'étranglement dans vos flux de travail, par exemple en décortiquant le délai nécessaire aux changements en étapes pour observer celles qui bloquent le processus. Lorsque les indicateurs DORA de vos équipes s'améliorent, l'efficacité de l'ensemble de la chaîne de valeur s'améliore avec elles.

En outre, les métriques DORA vous permettent de vous mesurer et vous améliorer pour ultimement :

- livrer plus rapidement, en garantissant une meilleure stabilité ;
- offrir plus de valeur aux clients en garantissant des itérations rapides et une meilleure réactivité au retour d'information ;
- augmenter plus rapidement la valeur commerciale d'un produit.



Équilibre entre vitesse et stabilité

Les mesures DORA donnent également un aperçu des performances des équipes. En examinant le taux d'échec des changements et le temps de récupération d'un échec de déploiement, les responsables de l'ingénierie peuvent s'assurer que leurs équipes construisent des produits robustes avec des temps d'arrêt minimaux. De même, le suivi de la fréquence de déploiement et du délai nécessaire aux changements permet d'avoir l'assurance que l'équipe travaille rapidement. Ensemble, ces mesures donnent un aperçu de l'équilibre entre vitesse et qualité au sein de l'équipe.

En comparant ces quatre indicateurs clés, on peut évaluer dans quelle mesure l'organisation parvient à équilibrer vitesse et stabilité. Par exemple, si le temps moyen de récupération d'un échec de déploiement est inférieur à une semaine avec des déploiements hebdomadaires, mais que le taux d'échec des changements est élevé, il se peut que l'équipe précipite les déploiements alors qu'il manque de

tests automatisés, ou elles peuvent ne pas être en mesure de prendre en charge les changements qu'elles déploient. D'autre part, si les déploiements ont lieu une fois par mois et que le temps de récupération d'un échec de déploiement et le taux d'échec des changements sont élevés, l'équipe passe peut-être plus de temps à corriger le code qu'à améliorer le produit. Tout est une question de contexte!

Les plus grands défis des métriques DORA

Accessibilité des données

La plupart du temps, les environnements et les données sont décentralisés, c'est-à-dire que les données sont dispersées dans différentes sources à travers l'organisation. De plus, les extraire peut être complexe lorsqu'elles ne sont disponibles qu'au format brut. Enfin, les données doivent être transformées et combinées en unités calculables pour permettre aux équipes de développement d'en tirer tous les avantages. En plus du temps requis pour faire ces manipulations, calculer manuellement les données laisse place à l'interprétation, ce qui présente un risque d'erreur ou de de communication d'informations erronées.

Analyse des résultats

Comme toutes les données, les mesures DORA doivent être mises en contexte et il faut considérer l'histoire que ces quatre métriques racontent ensemble. Le délai nécessaire aux changements et la fréquence de déploiement donnent un aperçu de la cadence d'une équipe et de la rapidité avec laquelle elle répond aux besoins en constante évolution des utilisateurs. D'autre part, le temps moyen de récupération d'un échec de déploiement et le taux d'échec des changements indiquent la stabilité d'un produit logiciel et la réactivité de l'équipe en cas d'interruption ou de défaillance du service. Seul un professionnel qui comprend la réalité de l'équipe de développement sera en mesure de bien interpréter ces données pour contribuer à leur amélioration continue.

Ce qu'il faut retenir

En réalité, les métriques ne sont qu'un outil. L'important, c'est que l'équipe de développement souhaite améliorer son efficacité et qu'elle utilise des indicateurs pour savoir si elle progresse dans ses efforts d'amélioration continue. Donnez à votre équipe les outils dont ils ont besoin pour réussir à apporter les meilleurs changements qui aideront votre équipe à atteindre ses objectifs.

Lorsque vos métriques DORA s'améliorent, vous pouvez être sûr que vous avez pris les bonnes décisions pour améliorer la performance de votre équipe et que vous apportez une plus grande valeur à vos clients.

Par où commencer?

Chez Axify, nous intégrons les métriques DORA à nos tableaux de bord pour voir tous les facteurs permettant de soutenir votre équipe dans l'amélioration continue d'un seul coup d'œil. Pas d'extraction de données ni d'ambiguïté sur leur interprétation. Tout est à portée de main pour rester dans le feu de l'action! De plus, puisque toutes vos équipes utiliseront la même structure et la même méthode d'extraction, il sera plus facile pour vous de comparer des pommes avec des pommes.

En savoir plus

Commencer



Comment implanter les métriques DORA dans votre processus?

Lorsque vous aurez choisi un outil vous permettant de suivre les métriques DORA, ne focalisez pas trop sur les métriques dans un premier temps. Commencez à collecter des données, suivez les métriques pendant quelques semaines, puis analysez ce que vous devez améliorer.

Ensuite, établissez des objectifs d'amélioration : concentrez-vous sur votre produit et sa croissance, ainsi que sur la croissance de votre équipe et l'amélioration des processus. Considérez les indicateurs DORA comme un ensemble d'indicateurs de ce que vous pouvez faire pour avoir un impact positif sur le produit et ses résultats commerciaux.

More Resources

- ✓ [Outil | Analyse de la maturité de l'équipe de développement](#)
- ✓ [Article de blogue | Le retour potentiel de l'amélioration continue en développement logiciel](#)
- ✓ [Article de blogue | Métriques de développement logiciel : pour vous fier à vos projections avec confiance](#)

Ability to measure team performance

66%

Communication and alignment between stakeholders

47%

Project predictability and planning

71%

À propos d'Axify



Connectez vos outils

Continuez à utiliser vos outils préférés comme Jira, Azure DevOps, GitHub et GitLab pour collecter des données sur les métriques DORA, la chaîne de valeur, le bien-être de votre équipe, les processus et plus.



Laissez Axify récupérer vos données

L'installation d'Axify prend moins de 10 minutes. Les données de l'ensemble du cycle de développement de votre logiciel sont disponibles en temps réel quelques minutes après la configuration. Adieu Excel et fichiers .csv !



Accédez à des performances élités

Obtenez toutes vos mesures d'ingénierie en un seul endroit pour prendre des décisions éclairées, aligner vos équipes de développement et améliorer vos processus de livraison logicielle.

Contactez-nous pour plus d'informations sur la façon dont Axify aide les équipes de développement à mesurer les indicateurs clés de performance DORA ou demandez une démonstration.

 info@axify.ca

 [Planifier une démo](#)

 +1 (418) 476-2090