

# ACCELERATE 2022

## Rapport sur l'état du DevOps

Sponsorié par



## Sommaire

<b>01</b>	<b>Note de synthèse</b>	03	<b>05</b>	<b>Résultats inattendus</b>	55
<b>02</b>	<b>Où en êtes-vous ?</b>	08	<b>06</b>	<b>Données sur les entreprises et démographiques</b>	59
<b>03</b>	<b>Comment s'améliorer ?</b>		<b>07</b>	<b>Conclusions</b>	67
	Introduction	19			
	Cloud	21	<b>08</b>	<b>Remerciements</b>	68
	Ingénierie SRE et DevOps	26			
	Capacités DevOps techniques	29	<b>09</b>	<b>Auteurs</b>	69
	Culture	37			
<b>04</b>	<b>L'importance de la sécurité sur la chaîne d'approvisionnement</b>	42	<b>10</b>	<b>Méthodologie</b>	73
			<b>11</b>	<b>Documentation complémentaire</b>	76

# 01

## Note de synthèse



Derek DeBellis



Claire Peters

Depuis huit ans, nous publions le rapport Accelerate sur l'état du DevOps, auquel ont déjà contribué 33 000 professionnels. Cette étude vise à analyser les indicateurs fonctionnels et stratégiques de certains domaines considérés comme essentiels dans le DevOps :

- Les performances en termes de livraison de logiciels, qui comprennent quatre indicateurs principaux : fréquence de déploiement, délai d'implémentation des modifications, taux d'échecs sur l'implémentation de modifications et délai de restauration du service.
- Les performances opérationnelles, évaluées à l'aide du cinquième indicateur principal, la fiabilité.
- Les performances organisationnelles, c'est-à-dire l'efficacité de votre organisation par rapport aux objectifs de performance et de rentabilité.

En outre, nous nous concentrons sur les facteurs sous-jacents à d'autres domaines, tels que le burnout et la probabilité que les employés recommandent leurs équipes.



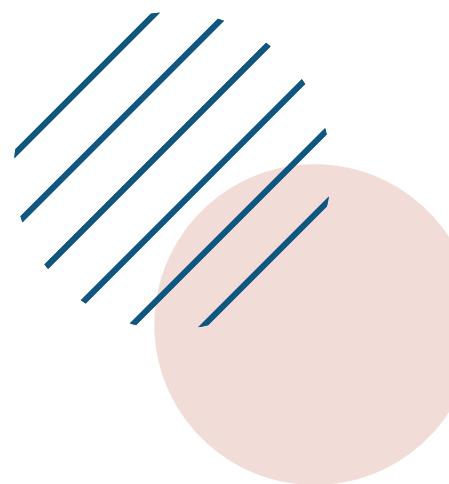
# Sécuriser la chaîne d'approvisionnement logicielle

En 2021, nous avons découvert qu'il était essentiel de sécuriser la chaîne d'approvisionnement logicielle pour atteindre de nombreux objectifs importants.

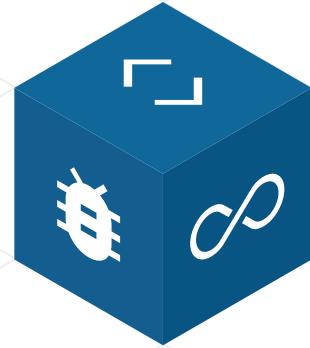
Cette année, nous avons étudié plus en détail cette sécurité, désormais un thème principal de notre enquête et de notre rapport. En nous appuyant sur le [framework SLSA](#) (Supply chain Levels for Software Artifacts, ou Niveaux de la chaîne d'approvisionnement pour les artefacts logiciels), nous avons exploré les stratégies techniques qui favorisent la sécurisation sur la chaîne d'approvisionnement logicielle. En outre, nous nous sommes basés sur le [NIST SSDF](#) (Secure Software Development Framework, ou Framework de développement de logiciels sécurisés du National Institute for Standards and Technology) pour explorer les conduites, les processus et les pratiques non techniques liés à la sécurisation sur la chaîne d'approvisionnement logicielle.

Selon notre étude, le principal indicateur des pratiques de sécurité des organisations pour le développement d'applications est culturel et non technique. Les cultures tournées vers les performances et non punitives sont 1,6 fois plus susceptibles d'adopter des stratégies orientées sécurité émergentes que les cultures à faible confiance et punitives, axées sur le pouvoir et les règles. Par ailleurs, nous avons identifié des signes précoce de l'efficacité des analyses de sécurité pré-déploiement dans la détection des dépendances vulnérables et donc la réduction du nombre de failles présentes dans le code en production.

L'adoption de bonnes pratiques de sécurité pour le développement d'applications présente d'autres avantages. Selon notre étude, lorsque les équipes se consacrent à la mise en œuvre de ces stratégies orientées sécurité, les risques de surmenage des développeurs sont réduits. Les équipes n'ayant qu'un faible recours à ces stratégies sont 1,4 fois plus susceptibles de présenter des niveaux importants de burnout que les équipes y ayant un recours élevé<sup>1</sup>. Les équipes qui privilégient la mise en place de pratiques de sécurité sont beaucoup plus à même de recommander leur équipe à des tiers. Plus précisément, l'utilisation de stratégies orientées sécurité liées au SLSA augurent de bonnes performances, tant organisationnelles qu'en termes de livraison de logiciels. Toutefois, cet impact nécessite de solides capacités d'intégration continue.



<sup>1</sup> Dans cette statistique, "élevé" correspond à un écart-type du score (ex. : sécurité)  $\geq 1$ , et "faible"  $\leq -1$ .



## Facteurs de performance organisationnelle

Outre les stratégies de sécurité mentionnées ci-dessus, les principaux vecteurs de performance organisationnelle correspondent généralement aux catégories suivantes :

### Culture organisationnelle et d'équipe

Les organisations qui favorisent une culture à niveau de confiance élevé et non punitive (telle que définie par [Westrum](#)) ont tendance à enregistrer de meilleures performances organisationnelles, de même que celles dont les équipes bénéficient de financements et du soutien des équipes dirigeantes. La stabilité et la perception positive d'une équipe (c'est-à-dire ses chances d'être recommandée) tendent également à favoriser les performances organisationnelles. Enfin, les entreprises qui offrent des conditions de travail flexibles enregistrent généralement des performances organisationnelles élevées.

### Fiabilité

Les pratiques associées à l'ingénierie de la fiabilité (objectifs de fiabilité clairs, métriques de fiabilité pertinentes, etc.), ainsi que la propension des équipes à déclarer atteindre leurs objectifs en la matière constituent des indicateurs importants des performances globales de l'entreprise.

### Cloud

Nous avons établi l'impact de l'utilisation du cloud sur les performances organisationnelles. Les entreprises dont les logiciels sont naturellement conçus avec et pour le cloud ont tendance à enregistrer des performances organisationnelles plus élevées. L'utilisation de clouds privés, publics ou hybrides est corrélée à des performances organisationnelles plus élevées que l'utilisation exclusive de serveurs sur site. Les organisations qui ont recours à plusieurs clouds publics sont 1,4 fois plus susceptibles de présenter des performances organisationnelles supérieures à la moyenne.

L'utilisation du cloud semble également avoir un impact sur les performances organisationnelles selon d'autres facteurs de notre ensemble de données. Par exemple, en ce qui concerne la sécurité sur la chaîne d'approvisionnement, nous avons constaté que les organisations qui utilisent des clouds publics sont également plus susceptibles de mettre en œuvre des pratiques de type SLSA. Cela peut être dû au fait que les fournisseurs de services cloud proposent des composants adaptés aux nombreuses pratiques SLSA, telles que l'automatisation des builds et des déploiements, et encouragent leur utilisation<sup>2,3</sup>. Plus généralement, on peut considérer que l'utilisation des plates-formes cloud permet à une équipe d'hériter de nombreuses capacités et pratiques, qui se traduisent par des performances organisationnelles supérieures.

<sup>2</sup> Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data", Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

<sup>3</sup> Carrión, Gabriel Cepeda, Christian Nitzl, and José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples". Partial least squares path modeling. Springer, Cham, 2017. 173-195.

## Tout dépend du contexte

Depuis longtemps, DORA tient compte de l'influence du contexte général des équipes sur les résultats. Nous sommes persuadés qu'il est important de comprendre les caractéristiques d'une équipe (processus, forces, contraintes et objectifs) et l'environnement dans lequel le travail a lieu. Par exemple, une même fonctionnalité technique peut s'avérer avantageuse ou délétère selon le contexte. Cette année, nous nous sommes concentrés sur la modélisation explicite de ces conditions hypothétiques sous la forme d'interactions. Les données de l'année en cours valident bon nombre de ces hypothèses.

- Un haut niveau de performance en termes de livraison de logiciels ne contribue aux performances organisationnelles que lorsque les performances opérationnelles sont également élevées. La rapidité de la livraison peut s'avérer inutile si votre service n'est pas en mesure de répondre aux attentes des utilisateurs concernant la fiabilité.
- La mise en œuvre de contrôles de sécurité sur la chaîne d'approvisionnement des logiciels, comme ceux recommandés par le framework SLSA, a un effet positif sur les performances de livraison de logiciels lorsque l'intégration continue est solidement établie. Sans capacités d'intégration continue en place, les performances de livraison de logiciels et les contrôles de sécurité peuvent entrer en conflit.
- L'impact des pratiques d'ingénierie en fiabilité des sites (SRE) sur la capacité d'une équipe à atteindre ses objectifs de fiabilité n'est pas linéaire. Les pratiques d'ingénierie SRE n'affectent pas positivement la fiabilité tant qu'une équipe n'a pas atteint un certain niveau de maturité dans cette discipline. Avant qu'une équipe atteigne ce niveau de maturité, nous n'observons pas de lien entre l'ingénierie SRE et les objectifs de fiabilité. Cependant, à mesure que l'équipe adopte l'ingénierie SRE, elle atteint un point d'infexion où le recours à l'ingénierie SRE devient un fort indicateur de fiabilité. L'amélioration de la fiabilité contribue alors aux performances organisationnelles.
- Les fonctionnalités techniques s'appuient les unes sur les autres. La livraison continue et le contrôle des versions amplifient mutuellement leur capacité à promouvoir des niveaux élevés de performances de livraison de logiciels. La combinaison de la livraison continue, d'une architecture faiblement couplée, du contrôle des versions et de l'intégration continue favorise des performances de livraison de logiciels supérieures à la somme de ses composants.



Les conditions dont dépend la livraison, et la nécessité de comprendre le contexte plus large d'une équipe, nous amènent à une conclusion semblable à cet insight de 2021 :

"Pour une amélioration significative des performances, les équipes doivent adopter une culture d'amélioration continue. Utilisez les benchmarks pour évaluer l'état actuel de votre organisation, identifier vos contraintes en fonction des capacités présentées dans le cadre de l'étude et tester des améliorations qui vous libèrent de ces contraintes. Lors de ces tests, vous rencontrerez à la fois des succès et des échecs. Dans les deux cas, vous en tirerez des enseignements qui permettront aux équipes de prendre des mesures efficaces."

En effet, nous avons constaté cette année un effet profondément aligné sur cette philosophie globale : les équipes qui reconnaissent la nécessité de s'améliorer continuellement ont tendance à présenter des performances organisationnelles supérieures à celles qui ne le font pas.

En bref, les équipes doivent s'adapter en permanence et tester les pratiques de développement logiciel.

En effet, les équipes qui adoptent un tel comportement présentent généralement des performances organisationnelles plus élevées, même si des exceptions existent (ce qui fonctionne pour une organisation ne fonctionne pas nécessairement pour une autre). Lorsque vous lancez dans vos propres expériences basées sur les pratiques DevOps, vous devez vous préparer à rencontrer des échecs occasionnels, et vous concentrer sur ce qui fonctionne pour votre équipe.

Cette année, les données ont également révélé quelques surprises que nous vous invitons à découvrir ci-après.

**Les équipes qui reconnaissent la nécessité de s'améliorer continuellement ont tendance à présenter des performances organisationnelles supérieures à celles qui ne le font pas.**

# 02

## Où en êtes-vous ?



Derek DeBellis

Souhaitez-vous comparer les performances de votre équipe à celles d'autres organisations du secteur ?

Cette section comprend la dernière évaluation de référence des performances DevOps. Nous observons comment les équipes développent, livrent et exploitent des systèmes logiciels. Nous répartissons ensuite les personnes interrogées en groupes qui représentent les combinaisons de performances DevOps les plus fréquemment rencontrées.

Cette année, cette répartition comprend deux approches de regroupement différentes. La première est tout simplement notre approche traditionnelle. Cette approche de regroupement vise à créer des groupes selon quatre indicateurs permettant de capturer les performances de livraison de logiciels : délai d'implémentation, fréquence de déploiement, délai de restauration du service et taux d'échecs sur l'implémentation de modifications. Nous en résumons les caractéristiques ci-après. Le but de cette approche est de vous aider à quantifier les performances actuelles de

votre équipe afin que vous puissiez les comparer à celles d'autres équipes.

La deuxième approche de regroupement comprend une cinquième métrique, la fiabilité, qui nous sert à comprendre les performances opérationnelles. Ce qui nous pousse à ajouter une nouvelle métrique à l'analyse par regroupement ? Nous avons constaté l'importance de cette métrique au fil du temps. En effet, nous avons déterminé que les performances de livraison peuvent nuire aux performances organisationnelles lorsqu'elles ne s'accompagnent pas de solides performances opérationnelles. Contrairement à notre approche de regroupement traditionnelle, cet exercice descriptif vise à brosser un portrait des performances communes des équipes en termes de livraison et d'opérations. Par conséquent, il n'est pas toujours évident de déterminer quel groupe est le meilleur.

Voyons d'abord un bref aperçu des cinq mesures sur lesquelles repose l'analyse des performances de livraison de logiciels et opérationnelles.



## Performances de livraison de logiciels et opérationnelles

Pour répondre aux besoins de leurs secteurs en constante évolution, les organisations doivent livrer et exploiter les logiciels de façon rapide et fiable. Plus vite vos équipes sont capables de modifier votre logiciel, plus tôt vous pourrez apporter de la valeur à vos clients, effectuer des tests et recevoir de précieux commentaires. En nous appuyant sur huit années de collecte de données et de recherche, nous avons développé et validé quatre métriques qui mesurent les performances de livraison de logiciels. Depuis 2018, nous avons inclus une cinquième métrique pour identifier les capacités d'exploitation. Les équipes

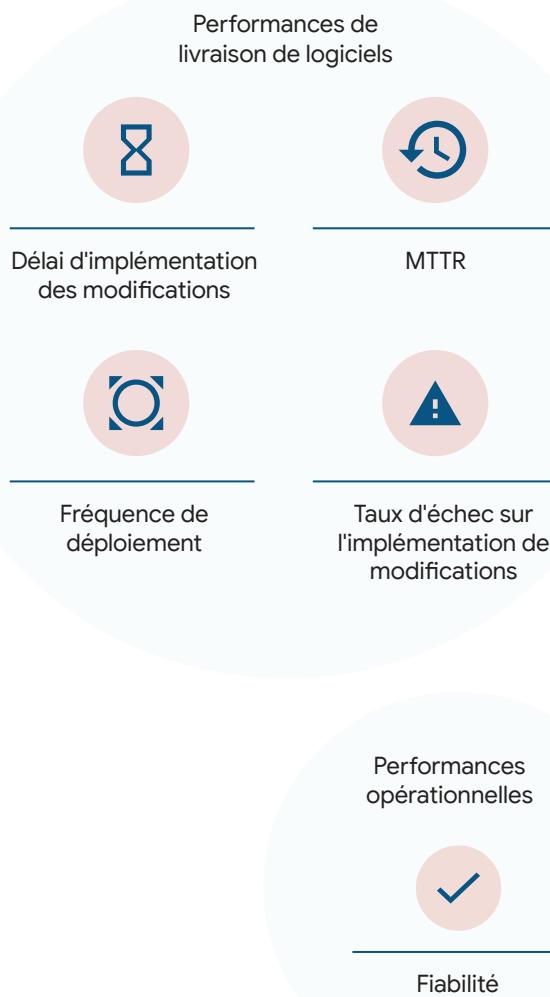
qui excellent dans l'ensemble de ces cinq indicateurs démontrent des performances organisationnelles exceptionnelles. Nous appelons ces cinq mesures : **performances de livraison de logiciels et opérationnelles (SDO)**. Veuillez noter que ces métriques concernent les objectifs au niveau du système, ce qui permet d'éviter les problèmes les plus courants du suivi des métriques logicielles qui peuvent entraîner l'opposition des fonctions les unes aux autres et procéder à des optimisations locales au détriment des résultats globaux.

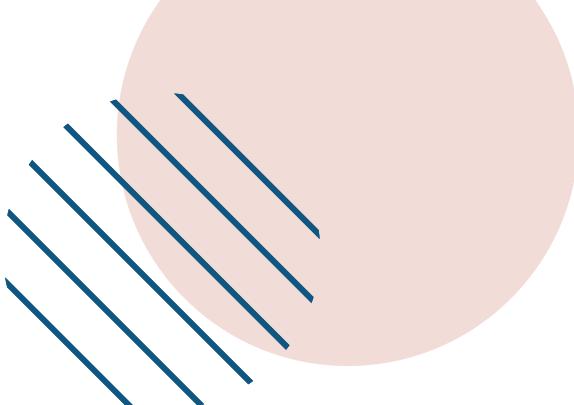


## Cinq métriques clés des performances opérationnelles et de livraison

Les quatre métriques évaluant les performances de la livraison de logiciels peuvent être envisagées sur le plan de la productivité et de la stabilité. Nous mesurons la productivité en étudiant le **délai d'implémentation des modifications du code** (c'est-à-dire, la durée entre la validation du code et la mise en production), et la **fréquence de déploiement**. Nous mesurons la stabilité grâce au **délai de restauration du service** après un incident et au **taux d'échecs sur l'implémentation de modifications**.

Une cinquième métrique représente les **performances opérationnelles** et permet de mesurer les pratiques opérationnelles modernes. Les **performances opérationnelles** s'appuient sur la **fiabilité**, c'est-à-dire le niveau de réponse de vos services aux attentes des utilisateurs, telles que la disponibilité et les performances. Nous mesurons traditionnellement la disponibilité plutôt que la fiabilité. Cependant, comme la disponibilité constitue un aspect spécifique de l'ingénierie SRE, nous avons étendu la mesure à la fiabilité en 2021 afin que la disponibilité, la latence, les performances et l'évolutivité soient plus largement représentées. Nous avons notamment demandé aux personnes interrogées d'évaluer leur capacité à atteindre ou dépasser leurs objectifs en termes de fiabilité. Nous avons constaté que les équipes dont les performances de livraison de logiciels varient enregistrent de meilleurs résultats (ex. : baisse du surmenage) lorsqu'elles privilégient également les performances opérationnelles.





## Approche de regroupement historique : rassembler les performances de livraison

Cette année, lors de l'évaluation de la solution à quatre groupes sur laquelle nous nous appuyons depuis 2018, nous avons remarqué que les données mettaient clairement en évidence un groupe à faible performance et un groupe à performance élevée. Cependant, les critères des deux groupes qui servaient traditionnellement à isoler les performances moyennes et les performances élevées n'étaient pas suffisamment différents pour justifier une division. Par ailleurs, les différents indicateurs sur lesquels repose le choix d'un groupe adapté suggèrent invariablement que trois

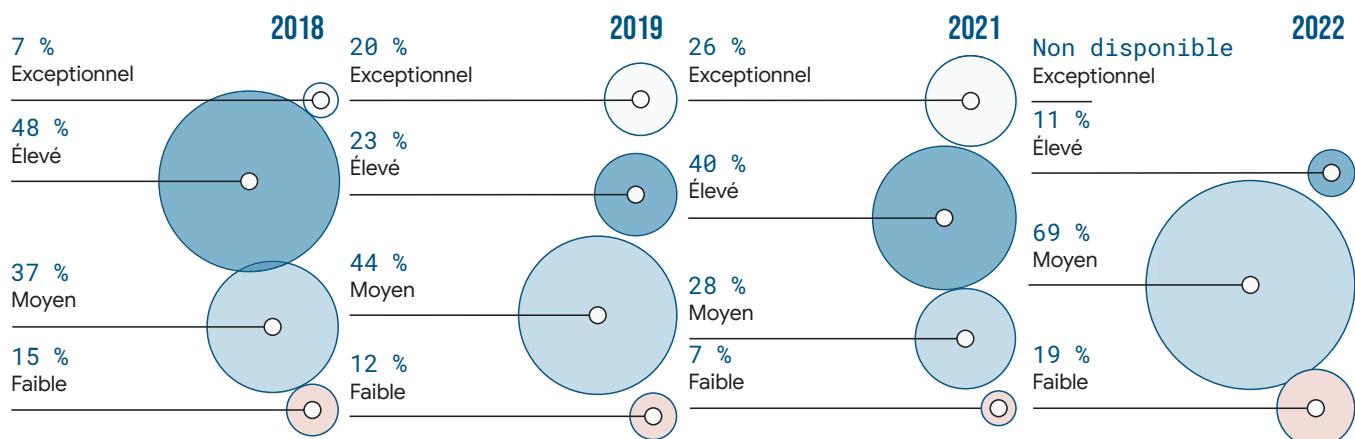
groupes permettent de capturer les données de manière optimale, quelle que soit la technique de regroupement appliquée. Le tableau ci-dessous décrit les caractéristiques des performances de livraison de chaque groupe.

La différence frappante par rapport à l'année dernière est que nous ne considérons désormais aucun groupe comme "exceptionnel". Cette année, le groupe de plus haut niveau combine des caractéristiques des groupes "élevé" et "exceptionnel". Nous avons décidé d'exclure le groupe "exceptionnel", car le groupe le plus performant de cette année ne présente tout simplement pas suffisamment de caractéristiques du groupe "exceptionnel" de l'année précédente.

Métrique de performances de livraison de logiciels	Faible	Moyen	Élevé
<b>Fréquence de déploiement</b> Pour l'application ou le service principal sur lequel vous travaillez, à quelle fréquence votre organisation déploie-t-elle du code en production ou le rend-t-elle accessible aux utilisateurs finaux ?	Entre une fois par mois et une fois tous les six mois	Entre une fois par semaine et une fois par mois	À la demande (plusieurs déploiements par jour)
<b>Délai d'implémentation des modifications</b> Pour l'application ou le service principal sur lequel vous travaillez, quel est votre délai d'implémentation des modifications (en d'autres termes, combien de temps faut-il pour passer du code validé au code en production) ?	Entre un mois et six mois	Entre une semaine et un mois	Entre un jour et une semaine
<b>Délai de restauration du service</b> En ce qui concerne l'application ou le service principal sur lequel vous travaillez, combien de temps faut-il pour restaurer le service en cas d'incident ou de défaut qui affecte les utilisateurs (par exemple, en cas de panne imprévue ou d'interruption du service) ?	Entre une semaine et un mois	Entre un jour et une semaine	Moins d'un jour
<b>Taux d'échec sur l'implémentation de modifications</b> En ce qui concerne l'application ou le service principal sur lequel vous travaillez, quel pourcentage de modifications en production ou mises à la disposition des utilisateurs entraîne une dégradation du service (par exemple, une interruption ou une panne du service) et nécessite par conséquent une correction (application de correctifs, rollback, fix forward, patch) ?	46 à 60 %	16 à 30 %	0 à 15 %

Cela suggère que cet échantillon ne représente pas des équipes ou des organisations avec des employés ayant l'impression d'avoir une longueur d'avance. Sans pouvoir valider cette hypothèse faute de données disponibles, nous supposons que cela indique une réduction de l'innovation en termes de développement de logiciels, notamment en ce qui concerne les pratiques, les outils et le partage d'informations. Il peut s'agir de l'une des conséquences de la pandémie actuelle, qui réduit les possibilités de partager les connaissances et les pratiques entre les équipes et les organisations. Les opportunités de se rassembler et d'apprendre les uns des autres se sont peut-être raréfiées, ce qui est susceptible d'avoir ralenti l'innovation. Nous espérons pouvoir approfondir nos recherches afin d'expliquer ce résultat.

Cela dit, en comparant les différents groupes (faible, moyen, élevé) de cette année avec ceux de l'année dernière, vous observerez une tendance légèrement à la hausse des performances de livraison. Il semble que les groupes de cette année se situent entre deux groupes de l'année dernière. En 2022, le groupe de haut niveau se situe entre le groupe "élevé" et le groupe "exceptionnel" de 2021. Quant au groupe de "faible", il semble se situer entre le groupe "faible" et le groupe "moyen" de 2021. La tendance haussière du groupe aux performances faibles suggère que si le niveau supérieur des performances de livraison baisse, le niveau inférieur augmente.

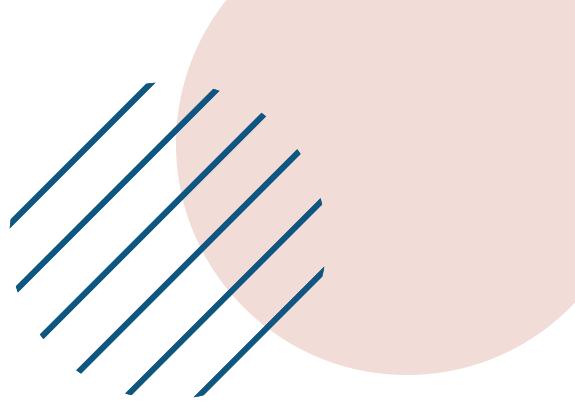


La répartition en pourcentage du tableau ci-dessus indique que la proportion d'entreprises très performantes est à son plus bas niveau depuis quatre ans, tandis que le pourcentage d'organisations peu performantes a considérablement augmenté, passant de 7 % en 2021 à 19 % cette année. Plus des deux tiers des personnes interrogées cette année appartiennent au groupe "moyen". La nette baisse du nombre d'entreprises très performantes (groupe "élevé" ou "exceptionnel") pourrait suggérer que bon nombre des personnes interrogées cette année font partie d'organisations ou d'équipes dont la mise en place d'une culture DevOps, qui émerge dans de nombreuses équipes modernes, n'a pas encore commencé ou est en cours.

Mais évitons de trop nous focaliser sur les différences entre 2021 et 2022, et mettons en évidence les ressemblances. Les groupes évoqués en 2021 et 2022 ont de nombreuses caractéristiques en commun, dont un fossé énorme entre les plus performants et les moins performants. Par exemple, on estime<sup>1</sup> que les entreprises les plus performantes effectuent 417 fois plus de déploiements que les entreprises les moins performantes.

<sup>1</sup> Voir la section "Méthodologie" pour en savoir plus sur cette estimation





## Regrouper les performances de livraison et les performances opérationnelles

Nous avons décidé d'effectuer une analyse des groupes sur les trois catégories suivantes (représentées par cinq métriques) : **productivité** (délai pour implémenter des modifications de code et fréquence des déploiements), **stabilité** (temps de restauration d'un service et taux d'échecs sur l'implémentation de modifications) et **performances opérationnelles** (fiabilité). Le choix de cette répartition a été motivé par le rôle central des performances opérationnelles

dans nos modèles. Pour les organisations qui ne présentent pas de solides performances opérationnelles, la productivité et la stabilité ont une influence moindre sur les performances organisationnelles. Nous pensons qu'une description du paysage des performances en DevOps qui laisse de côté les performances opérationnelles ne donne qu'une vision partielle des choses.

L'exploration des données nous a conduits à une solution à quatre groupes. Voici le détail et le nom des quatre groupes :

Groupe	Stabilité		Performances opérationnelles	Productivité		% des personnes interrogées
	Délai de restauration du service	Taux d'échec sur l'implémentation de modifications	Fiabilité	Délai d'implémentation	Fréquence de déploiement	
En rodage	Entre un jour et une semaine	31 à 45 %	Répond parfois aux attentes	Entre une semaine et un mois	Entre une fois par semaine et une fois par mois	28 %
Dynamique	Moins d'une heure	0 à 15 %	Répond généralement aux attentes	Moins d'un jour	À la demande (plusieurs déploiements par jour)	17 %
En décélération	Moins d'un jour	0 à 15 %	Répond généralement aux attentes	Entre une semaine et un mois	Entre une fois par semaine et une fois par mois	34 %
En retrait	Entre un mois et six mois	46 à 60 %	Répond généralement aux attentes	Entre un mois et six mois	Entre une fois par mois et une fois tous les six mois	21 %

Toutefois, si vous rendez visite à deux équipes du même groupe, il est probable qu'elles vous semblent très différentes. Notre présentation risque de ne pas refléter vos impressions. Nous présentons chaque groupe ci-dessus de manière à exploiter notre expérience de travail avec de nombreuses équipes pour rendre compréhensibles ces modèles de données. En outre, si vous rendez visite à la même équipe à deux moments différents, il est possible qu'elle ait changé de groupe entre temps. Ce changement peut être dû à une amélioration ou à une baisse du niveau de l'équipe, ou encore à un changement de son modèle de déploiement pour une solution plus adaptée à l'état actuel de son application ou de son service. Par exemple, à un stade antérieur du développement d'une application ou d'un service, l'équipe s'est focalisée sur l'exploration (groupe "En rodage"), mais a déplacé le curseur sur la fiabilité (groupe "Dynamique" ou groupe "En décélération") à mesure qu'elle trouvait son créneau.

## Chaque groupe présente des caractéristiques uniques et représente une proportion importante des réponses.

Le groupe **En rodage** présente des performances ni bonnes ni mauvaises pour l'ensemble des dimensions. Ce groupe peut en être aux premiers stades du développement de son produit, de sa fonctionnalité ou de son service. Il peut considérer la fiabilité comme non prioritaire et se focaliser sur le recueil de retours, l'analyse de l'adéquation du produit au marché et, plus généralement, l'exploration.

Le groupe **Dynamique** présente de hautes performances pour l'ensemble des caractéristiques (fiabilité, stabilité et productivité élevées). Seuls 17 % des personnes interrogées font partie de ce groupe.

Les entreprises du groupe **En décélération** n'effectuent pas souvent de déploiements, mais sont susceptibles de les réussir. Plus d'un tiers des personnes interrogées appartiennent à ce groupe, ce qui en fait le plus important et le plus représentatif de notre échantillon. Ce modèle est probablement typique (bien que loin d'être exclusif) des équipes qui s'améliorent progressivement, mais qui sont généralement satisfaites de l'état actuel de leur application ou produit, tout comme leurs clients.

Pour finir, le groupe **En retrait** rassemble des équipes travaillant sur des services ou des applications qui ne sont plus en cours de développement actif, mais qui n'en sont pas moins valorisées par elles-mêmes et par leurs clients.

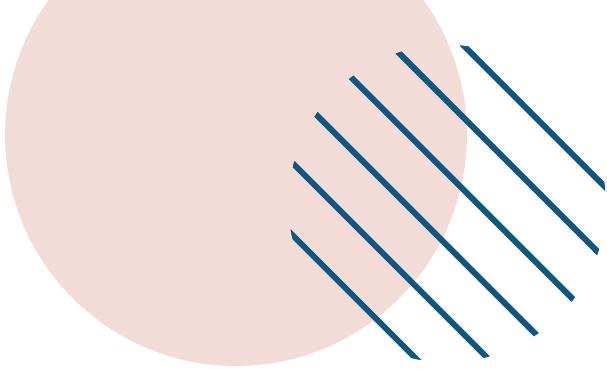
Ces groupes présentent des pratiques et des capacités techniques très variées. Compte tenu des hautes performances de livraison de logiciels et opérationnelles du groupe **Dynamique**, nous avons décidé d'examiner en quoi leurs pratiques et leurs capacités techniques diffèrent des autres groupes. Selon nos recherches, le groupe **Dynamique** met davantage l'accent sur certaines caractéristiques :

- Architectures faiblement couplées : capacité des équipes à modifier à grande échelle la conception de leur système sans dépendre d'autres équipes
- Souplesse : degré de flexibilité d'une entreprise en ce qui concerne les conditions de travail des employés
- Contrôle des versions : manière de gérer des modifications apportées au code d'application, à la configuration du système, à la configuration de l'application, etc.
- Intégration continue (CI) : fréquence d'intégration des branches dans la branche principale
- Livraison continue (CD) : capacités axées sur la mise en production des modifications de manière sûre, durable et efficace

Curieusement, le groupe **Dynamique** a tendance à considérer la documentation comme moins prioritaire. L'année dernière, nous avions constaté que les pratiques de documentation étaient essentielles aux performances de livraison et aux performances opérationnelles (SDO). Comment se fait-il que le groupe **Dynamique** présente des performances SDO élevées sans se focaliser sur la documentation ? D'une part, la documentation n'est pas le seul moyen d'obtenir de bonnes performances SDO. D'autre part, il est possible que le groupe **Dynamique** refactorise continuellement son code afin d'encourager l'autodocumentation, ce qui réduit la quantité de documents nécessaires tels que nous les décrivons dans l'enquête.

Le groupe **En décélération**, qui rassemble la majorité des personnes interrogées, se compose généralement de grandes organisations naturellement moins cloud natives que les autres groupes. Prenez le cas d'une entreprise très mature s'appuyant sur des processus endurcis qui, en fin de compte, offre toujours aux utilisateurs finaux des expériences stables, fiables et valorisées. Ce groupe présente une culture génératrice axée sur les performances<sup>2</sup>. L'une des caractéristiques les plus intéressantes du groupe En décélération est que sa productivité est faible, mais sa culture d'entreprise très positive ("générative" selon Westrum). C'est une combinaison peu commune. Il est plus courant que la productivité soit proportionnelle à la culture d'entreprise (rapport élevé/élevé ou faible/faible). Nous espérons que nos prochaines recherches permettront de mieux comprendre la relation entre la productivité et la culture d'une entreprise.

<sup>2</sup> (Westrum)



Nous avons également comparé les différents groupes selon trois domaines : le burnout, les performances organisationnelles et les tâches non planifiées. Le résultat de nos recherches nous a surpris : les performances organisationnelles du groupe **En retrait** dépassent celles des autres groupes. En observant les caractéristiques de ce groupe (stabilité et productivité faibles), ce résultat contredit la plupart des conclusions précédentes de DORA. S'il est fort possible que celui-ci soit dû au hasard, nous souhaitons plutôt explorer quelques pistes d'explication.

Lors de l'analyse des conclusions, un résultat complémentaire important est à considérer. Le groupe **En retrait** atteint des performances organisationnelles élevées à un coût tout aussi élevé : ses équipes présentent les taux de burnout les plus importants, se sentent plus susceptibles de commettre des erreurs et sont davantage accablées par les tâches imprévues. En parallèle, ces résultats suggèrent que si le critère de fiabilité peut suffire pour atteindre des performances organisationnelles élevées, la rapidité et la stabilité restent essentielles pour éviter le surmenage des équipes et les tâches imprévues.

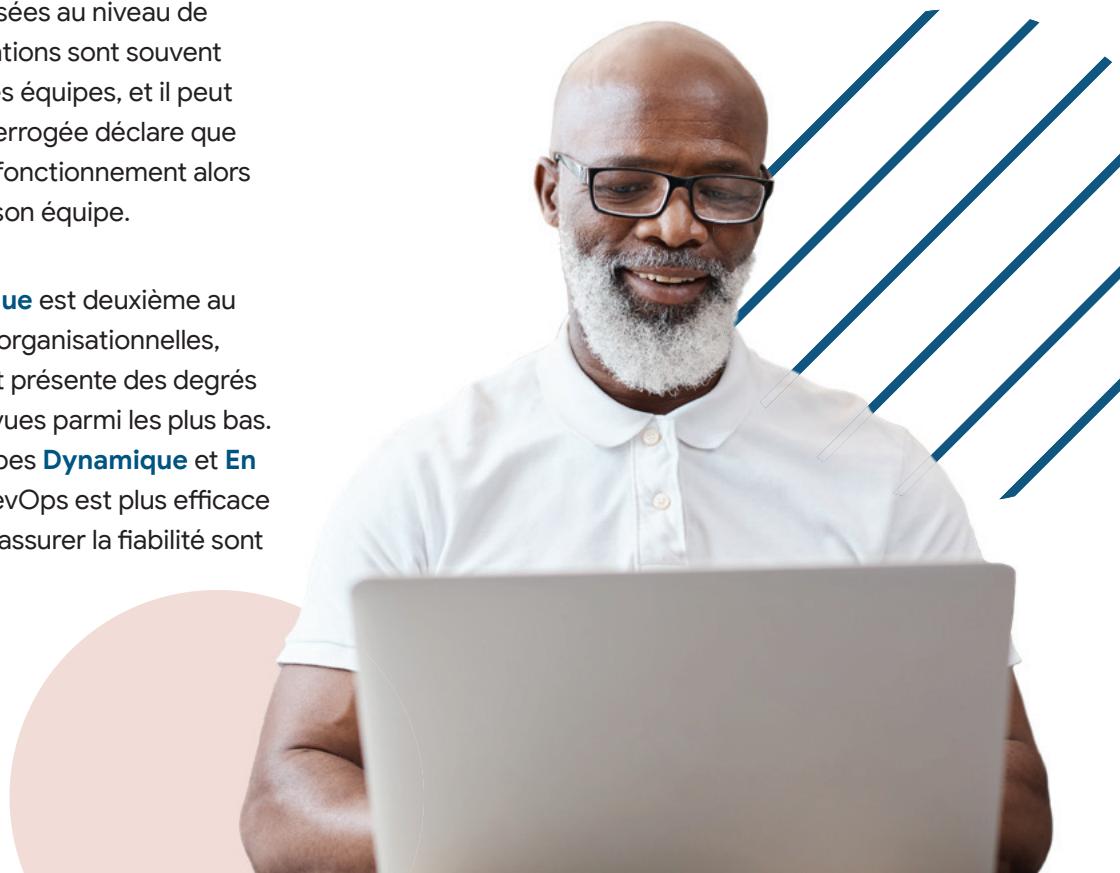
Nous avons d'autres hypothèses pour expliquer pourquoi les performances organisationnelles du groupe **En retrait** dépassent celles des autres groupes, et plus particulièrement celles du groupe **Dynamique**. Les voici.

- Certaines caractéristiques sous-jacentes à ces quatre groupes peuvent ne pas figurer dans nos données. Par exemple, la taille de l'organisation peut être un bon indicateur de maturité. Le groupe **Dynamique** est composé d'entreprises généralement plus petites, ce qui peut indiquer que leurs produits sont à des stades moins avancés.
- Les membres du groupe **Dynamique** sont généralement issus de plus petites entreprises, qui peuvent être moins dépendantes de processus et d'infrastructures historiques et, par conséquent, avoir mis en place des processus DevOps plus sophistiqués. Cela dit, les données montrent que la taille d'une organisation est positivement corrélée aux performances organisationnelles pour des raisons pouvant être largement indépendantes de la technologie.
- Le groupe **Dynamique** a tendance à rejeter les principes décrits dans la culture générative de Westrum. Comme nous l'avons vu précédemment, ce rejet se fait souvent au détriment des performances organisationnelles.
- Les organisations appartenant à chaque groupe peuvent avoir une conception différente des attentes en termes de fiabilité et des méthodes permettant de la surveiller. Il en va de même concernant la définition de leurs objectifs en termes de performances organisationnelles.

- Le groupe **En retrait** peut présenter des performances organisationnelles élevées à court terme, sans que celles-ci ne soient garanties à long terme. Le burnout se traduit-il par un renouvellement important du personnel ? Les processus peuvent-ils évoluer ?
- Les questions sont posées à des niveaux différents. Concernant les capacités techniques (une architecture faiblement couplée, par exemple), c'est l'équipe qui est interrogée, tandis que les questions de performances organisationnelles sont posées au niveau de l'organisation. Les organisations sont souvent composées de nombreuses équipes, et il peut arriver que la personne interrogée déclare que son organisation a un bon fonctionnement alors que ce n'est pas le cas de son équipe.

Par ailleurs, le groupe **Dynamique** est deuxième au classement des performances organisationnelles, derrière le groupe **En retrait**, et présente des degrés de burnout et de tâches imprévues parmi les plus bas. Comme observé dans les groupes **Dynamique** et **En décélération**, la philosophie DevOps est plus efficace lorsque des méthodes visant à assurer la fiabilité sont en place.

Nous avons hâte de poursuivre nos recherches afin de découvrir de nouvelles façons de décrire les différences qui existent dans le secteur. À l'avenir, nous souhaitons continuer à considérer les performances opérationnelles comme une dimension pertinente de notre analyse de ces différences. Nous voulons également éviter de suggérer des groupes trop normatifs et évaluatifs (par exemple, "exceptionnel") et nous concentrer plutôt sur l'identification et la description pure et simple des configurations communes en termes de performances SDO.



# 03

## Comment s'améliorer ?



Derek DeBellis

### Comment s'améliorer dans de nombreux domaines ?

Le rapport sur l'état du DevOps est conçu pour fournir des conseils étayés par des faits, afin d'aider votre équipe à se concentrer sur les pratiques et capacités DevOps qui permettent d'obtenir des résultats pertinents. Cette année, nous avons étendu le spectre de notre enquête à la sécurité et à l'ensemble des résultats attendus par les équipes. Dans le passé, nous nous concentrons sur les performances de livraison de logiciels et opérationnelles (SDO) ainsi que sur les effets en termes de performances organisationnelles. C'est toujours le cas, mais nous souhaitons également explorer le burnout, la probabilité de recommandation des équipes, les tâches imprévues et la tendance à l'erreur, non seulement comme facteurs d'amélioration des performances SDO et organisationnelles, mais aussi comme résultats en tant que tels. Par conséquent, cette année, nous avons veillé à lister les pratiques et les capacités qui semblent avoir une influence sur ces résultats.





Le modèle de recherche a évolué en 2022 afin de mieux refléter une théorie sous-jacente de DORA : il n'existe pas d'approche unique du DevOps. En pratique, nous avons constaté que pour formuler des recommandations, il faut assimiler le contexte global d'une équipe. Une pratique bénéfique pour une équipe donnée peut s'avérer préjudiciable pour une autre équipe. Par exemple, nous avons longtemps émis l'hypothèse que les capacités techniques (telles que l'architecture faiblement couplée, le développement à branche unique, le contrôle des versions et l'intégration continue) ont un impact positif plus prononcé sur les performances de livraison de logiciels lorsque la livraison continue est en place. Cette année, nous avons explicitement modélisé cette interaction, parmi d'autres. L'objectif est d'améliorer notre analyse en ne nous demandant plus seulement ce qui a un effet sur quoi, mais aussi dans quelles conditions ces effets peuvent se produire, s'amplifier ou s'atténuer. L'analyse de cet ensemble de conditions s'est avérée incroyablement complexe, mais nous sommes ravis de vous présenter un aperçu de ces premiers résultats.

Vous retrouverez les modèles de recherche utilisés cette année et les années précédentes sur notre [site Web](#).

## Au-delà des quatre indicateurs

Comment les métriques DORA améliorent-elles les performances du développement et des opérations ? Une équipe pluridisciplinaire d'ingénieurs logiciels chez Liberty Mutual Insurance évalue régulièrement ses performances en s'appuyant sur les [quatre indicateurs de DORA](#). Par exemple, Jenna Dailey, Scrum Master senior chez Liberty Mutual, explique que les études de DORA ont permis à l'équipe d'identifier un goulot d'étranglement, d'adopter une approche de développement piloté par les tests et d'améliorer ses performances globales.

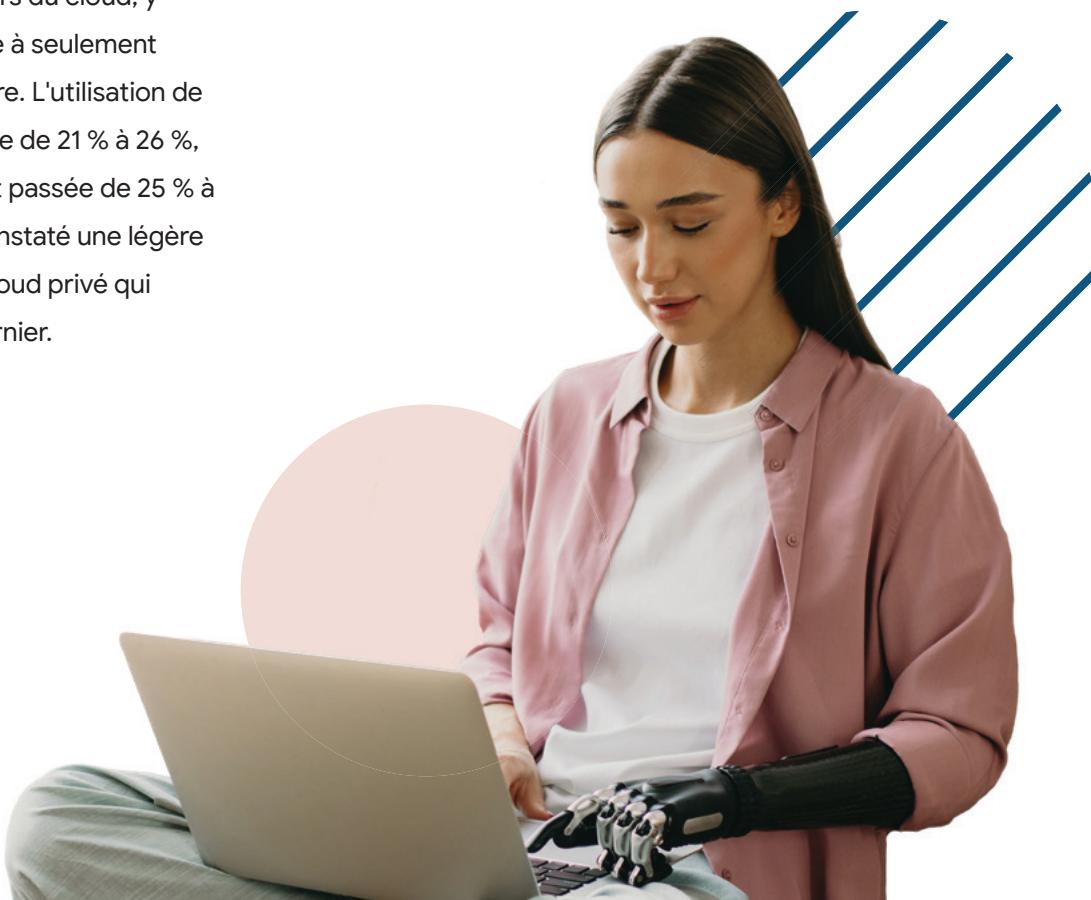
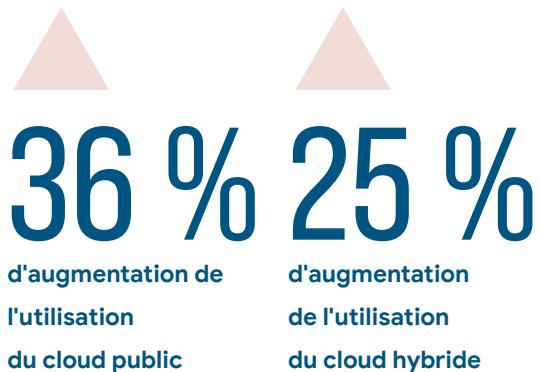
Découvrez comment Liberty Mutual a exploité les données et les métriques DORA pour améliorer la qualité et la livraison de ses logiciels dans sa dernière conférence [Tomorrow Talks](#).



Eric Maxwell

## Cloud

Dans la continuité des tendances observées ces dernières années, l'utilisation du cloud computing continue de se développer. De fait, le pourcentage d'utilisateurs du cloud public, y compris de plusieurs clouds, s'élève désormais à 76 %, contre 56 % en 2021. Le nombre de non-utilisateurs du cloud, y compris du cloud privé, est tombé à seulement 10,5 %, contre 21 % l'année dernière. L'utilisation de plusieurs clouds publics est passée de 21 % à 26 %, et l'utilisation du cloud hybride est passée de 25 % à 42,5 %. Nous avons également constaté une légère augmentation de l'utilisation du cloud privé qui atteint 32,5 %, contre 29 % l'an dernier.



L'utilisation du cloud computing a un impact positif sur les performances organisationnelles globales. Les personnes interrogées qui utilisent le cloud sont **14 % plus susceptibles de dépasser** les objectifs de performances organisationnelles que celles qui s'en passent.

Comme nous l'avons montré au cours des années précédentes, et nous le validons à nouveau dans ce rapport, l'utilisation du cloud computing a un impact positif sur les performances organisationnelles globales. Les personnes interrogées qui utilisent le cloud sont **14 % plus susceptibles de dépasser** les objectifs de performances organisationnelles que celles qui s'en passent. Selon nos recherches, le cloud computing permet aux équipes d'exceller dans des domaines tels que la sécurité et la fiabilité sur la chaîne d'approvisionnement logicielle, qui sont des facteurs de performances organisationnelles.

Étonnamment, quel que soit le type de cloud utilisé (public, privé, hybride et multicloud), l'association avec le taux d'échecs sur l'implémentation de modifications est négative. En d'autres termes, ce taux a augmenté.

Cela nécessite un examen plus approfondi. Au lieu de spéculer sur les causes de cette augmentation, nous approfondirons nos recherches à l'occasion d'une prochaine étude. Toutefois, à quelques exceptions près, l'utilisation d'applications cloud natives (c'est-à-dire initialement conçues et architecturées pour le cloud) a montré des signaux positifs dans tous les domaines étudiés.

L'utilisation de toute plate-forme de cloud computing, publique ou privée, contribue positivement aux résultats en termes de culture d'entreprise et d'environnement de travail (culture générative, réduction du burnout, meilleure stabilité, et satisfaction des employés accrue, etc.). Les utilisateurs du cloud ont enregistré des résultats 16 % plus élevés en termes de culture d'entreprise.

## L'utilisation du cloud continue d'augmenter d'une année sur l'autre

	2022	Évolution en % par rapport à 2021
Cloud hybride	42,47 %	25 %
Cloud public et/ou plusieurs clouds publics	76,08 %	36 %
Cloud privé	32,55 %	12 %
Aucun cloud	10,55 %	-50 %



L'utilisation du cloud hybride et du multicloud (ainsi que du cloud privé) semble avoir un impact **négatif** sur plusieurs indicateurs de performances de livraison de logiciels (MTTR, délai et fréquence de déploiement) **sauf** lorsque les personnes interrogées signalent des niveaux élevés de **fiabilité**.

## Le cloud hybride et le multicloud stimulent les performances organisationnelles

Nous continuons d'observer des signaux forts indiquant que l'utilisation du cloud hybride et de plusieurs clouds publics a un impact positif sur les organisations. Les professionnels qui utilisent plusieurs clouds font preuve de performances organisationnelles 1,4 fois supérieures à celle des non-utilisateurs du cloud. L'utilisation du cloud hybride et du multicloud (tout comme du cloud privé) semble avoir un impact négatif sur plusieurs indicateurs de performances de livraison de logiciels (MTTR, délai et fréquence de déploiement), sauf lorsque les personnes interrogées signalent des niveaux de fiabilité élevés. Ce résultat

témoigne de l'importance d'une pratique d'ingénierie SRE robuste et du rôle de la fiabilité dans la livraison de logiciels.

En 2021, nous avions demandé aux personnes interrogées de nous indiquer leur motivation *principale* pour utiliser plusieurs clouds publics, tandis qu'en 2022, nous leur avons demandé de citer l'ensemble des avantages qu'elles tiraient de l'utilisation de plusieurs fournisseurs de services cloud. La disponibilité est l'avantage le plus cité, ce qui coïncide avec l'attention particulière portée sur la fiabilité dans le secteur : pour être fiables, les services doivent avant tout être disponibles. Plus de 50 % des professionnels déclarent exploiter les avantages propres aux différents fournisseurs de services cloud.

## Bénéfices de l'adoption de plusieurs fournisseurs de services cloud

Disponibilité	62,61 %
Exploitation des avantages uniques de chaque fournisseur	51,59 %
La confiance est répartie entre plusieurs fournisseurs	47,54 %
Reprise après sinistre	43,48 %
Conformité réglementaire	37,97 %
Technique de négociation ou exigence pour le provisionnement	19,13 %
Autre	4,06 %

> 50 %

des personnes interrogées déclarent utiliser plusieurs fournisseurs de services cloud.

Le recours aux cinq caractéristiques du cloud computing est le point de départ incontournable d'une longue chaîne causale vers les performances organisationnelles.

## Les cinq caractéristiques du cloud computing

Conformément à l'approche utilisée lors de notre étude précédente, nous avons cherché à savoir si les participants utilisent des technologies de cloud computing, mais aussi comment ils les utilisent. Nous y sommes parvenus en nous interrogeant sur les cinq caractéristiques essentielles du cloud computing, telles que définies par le National Institute of Standards and Technology (NIST).

**Libre service à la demande** : les consommateurs peuvent provisionner automatiquement des ressources informatiques selon les besoins, sans aucune intervention humaine nécessaire de la part du fournisseur.

**Accès réseau étendu** : les fonctionnalités sont largement disponibles et les consommateurs peuvent y accéder via plusieurs clients (téléphones mobiles, tablettes, ordinateurs portables, stations de travail, etc.).

**Pooling de ressources** : les ressources du fournisseur sont regroupées dans un modèle mutualisé, et les ressources physiques et virtuelles sont attribuées dynamiquement à la demande.

Le client n'a généralement aucun contrôle direct sur l'emplacement exact des ressources fournies, mais il peut spécifier l'emplacement à un niveau d'abstraction plus élevé, comme le pays, l'état ou le centre de données.

**Élasticité rapide** : les fonctionnalités peuvent être provisionnées et libérées de manière flexible afin d'évoluer rapidement à la hausse ou à la baisse selon la demande. Les fonctionnalités consommateur disponibles au provisionnement sont virtuellement illimitées et leur quantité peut être ajustée à tout moment.

**Service mesuré** : les systèmes cloud contrôlent et optimisent automatiquement l'utilisation des ressources en exploitant une capacité de mesure à un niveau d'abstraction approprié au type de service, comme le stockage, le traitement, la bande passante et les comptes utilisateur actifs. Pour plus de transparence, l'utilisation des ressources peut être surveillée, contrôlée et signalée.

Ce rapport confirme les études DORA des trois années précédentes, et conclut que la présence de ces cinq caractéristiques dans une organisation a un impact positif sur les performances opérationnelles et de livraison de logiciels. Nous avons également constaté que ces caractéristiques conduisent à une meilleure performance organisationnelle, car elles mobilisent des processus bénéfiques pour l'organisation.

Présenter les cinq caractéristiques du cloud computing constitue le point de départ d'un long parcours vers de meilleures performances organisationnelles.

En 2022, nous constatons que les équipes profitent de plus en plus des caractéristiques du cloud computing. Pour la quatrième année consécutive, nous constatons une adoption croissante des cinq caractéristiques du cloud computing. Le pooling de ressources a enregistré la plus forte augmentation (+ 14 %), tandis que l'élasticité rapide, qui était la deuxième fonctionnalité la plus utilisée l'année dernière, a enregistré la plus faible augmentation (+ 5 %).

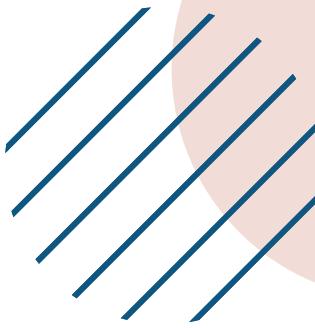


NIST	2021	2022	Variation en pourcentage
Accès réseau étendu	74 %	80 %	8
Élasticité rapide	77 %	81 %	5
Libre-service à la demande	73 %	78 %	7
Service sur mesure	78 %	83 %	7
Pooling des ressources	73 %	83 %	14

  
**14 %**  
d'augmentation  
du pooling de  
ressources



Dave Stanke



## Ingénierie SRE et DevOps

Les équipes technologiques performantes font bien plus pour leurs organisations que livrer du code, même lorsque ce code est de qualité. Elles s'assurent également que les services qu'elles fournissent restent disponibles, performants et conformes aux attentes des utilisateurs au fil du temps. La fiabilité est une mesure à multiples facettes de la capacité d'une équipe à respecter ces engagements. Cette année, nous avons approfondi nos recherches concernant la fiabilité en tant que facteur de livraison et d'exploitation de logiciels.

L'ingénierie en fiabilité des sites (SRE) est une approche influente des opérations qui a vu le jour chez Google et est désormais pratiquée dans de nombreuses organisations. L'ingénierie SRE donne la priorité à l'apprentissage empirique, à la coopération pluridisciplinaire, au recours étendu à l'automatisation, et à l'utilisation de techniques de mesure comme les SLO (objectifs de niveau de service). D'autres pratiques opérationnelles modernes utilisent des méthodes semblables, mais appliquent des conventions de nommage différentes. Par conséquent, afin d'évaluer le plus objectivement possible l'étendue de ces pratiques, nous avons pris soin d'utiliser un langage neutre et descriptif dans le texte de l'enquête que nous présentons

aux personnes interrogées. Nous recueillons également des données sur les résultats de l'ingénierie en fiabilité, c'est-à-dire la propension des équipes à atteindre leurs objectifs de fiabilité. Les entrées et les sorties (pratiques d'ingénierie SRE et résultats en termes de fiabilité) sont représentées dans notre modèle prédictif aux côtés d'autres capacités DevOps.

### La fiabilité est primordiale

L'adoption de l'ingénierie SRE est répandue parmi les équipes que nous avons interrogées : la majorité des personnes interrogées déclarent avoir recours à au moins une des pratiques abordées dans le questionnaire. Dans cette diversité d'équipes, les données révèlent une relation nuancée entre la fiabilité, la livraison de logiciels et les résultats : lorsque la fiabilité est faible, les performances de livraison de logiciels ne sont pas un indicateur de réussite organisationnelle. Cependant, lorsque la fiabilité est plus élevée, nous commençons à percevoir l'influence positive de la livraison de logiciels sur la réussite de l'entreprise.

**Sans fiabilité, les performances de livraison de logiciels ne sont pas un indicateur de réussite organisationnelle.**

## L'investissement dans l'ingénierie SRE permet d'améliorer la fiabilité, mais seulement une fois qu'un seuil d'adoption a été atteint.

Ce phénomène est cohérent avec l'utilisation du framework "marge d'erreur" de l'ingénierie SRE : lorsqu'un service n'est pas fiable, le contexte est fragile, et les utilisateurs ne bénéficieront pas d'une mise en production de code plus rapide.

Comme l'affirment depuis longtemps les ingénieurs en fiabilité des sites, la fiabilité est la "fonctionnalité" la plus importante de tout produit. Nos recherches confirment l'observation selon laquelle tenir les promesses faites aux utilisateurs est une condition sine qua non pour que l'amélioration de la livraison de logiciels profite à l'organisation.

## Reconnaitre la courbe en J

Quels défis vous attendent dans votre recherche de fiabilité ? Dans "Enterprise Roadmap to SRE"<sup>1</sup>, publié chez O'Reilly, James Brookbank et Steve McGhee, contributeurs de l'enquête DORA, reviennent sur leur expérience de mise en œuvre de l'ingénierie SRE dans des organisations établies et recommandent de "reconnaitre la courbe en J du changement". Le phénomène de la courbe en J, abordé dans le rapport sur l'état du DevOps en 2018, décrit la tendance des transformations organisationnelles à générer des succès rapides suivis de périodes de rendements décroissants,

voire de régressions. Cependant, les organisations qui relèvent malgré tout ces défis connaissent souvent un niveau de réussite élevée, répétée et durable.

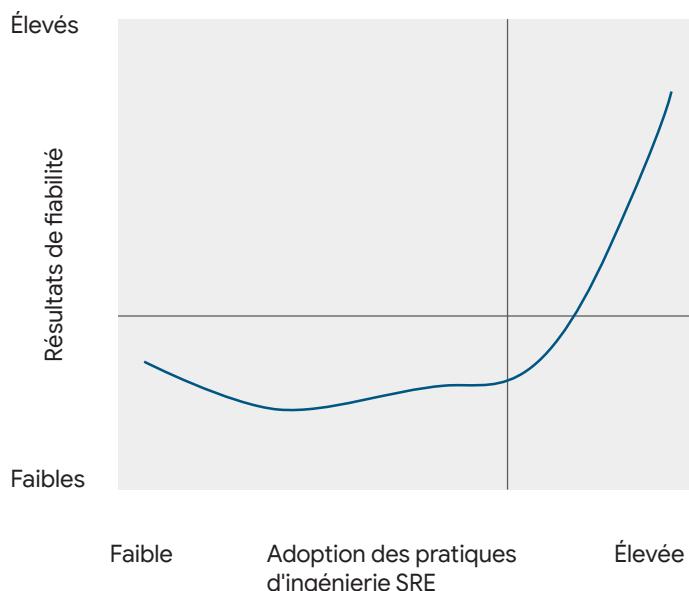
Notre étude 2022 met en évidence un modèle de courbe en J dans les équipes techniques interrogées : lorsqu'elles s'appuient sur moins de pratiques en ingénierie de fiabilité (ce qui suggère qu'elles n'en sont qu'au début de leur parcours d'adoption de l'ingénierie SRE), ces pratiques n'indiquent pas de meilleurs résultats en termes de fiabilité. En revanche, à mesure que les équipes adoptent davantage de pratiques d'ingénierie SRE, elles atteignent un point d'inflexion où le recours aux pratiques devient un fort indicateur de la fiabilité, puis, par conséquent, des performances organisationnelles.



<sup>1</sup> <https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/>

## Les équipes fiables rendent les services fiables : la culture d'équipe génératrice est un indicateur d'une plus grande fiabilité.

Les équipes qui débutent leur parcours d'adoption des pratiques d'ingénierie SRE doivent s'attendre à faire face à des épreuves en cours de route. Ce parcours peut prendre du temps, car la culture, les processus et les outils doivent tous s'adapter aux nouveaux principes directeurs. Que les équipes se rassurent : au fil du temps et grâce à un investissement continu, la réussite est probable.



Les équipes qui poursuivent leurs efforts au-delà des étapes initiales de l'adoption de l'ingénierie SRE constatent une amélioration croissante de la fiabilité

## Mettre l'accent sur les équipes, les processus et les outils

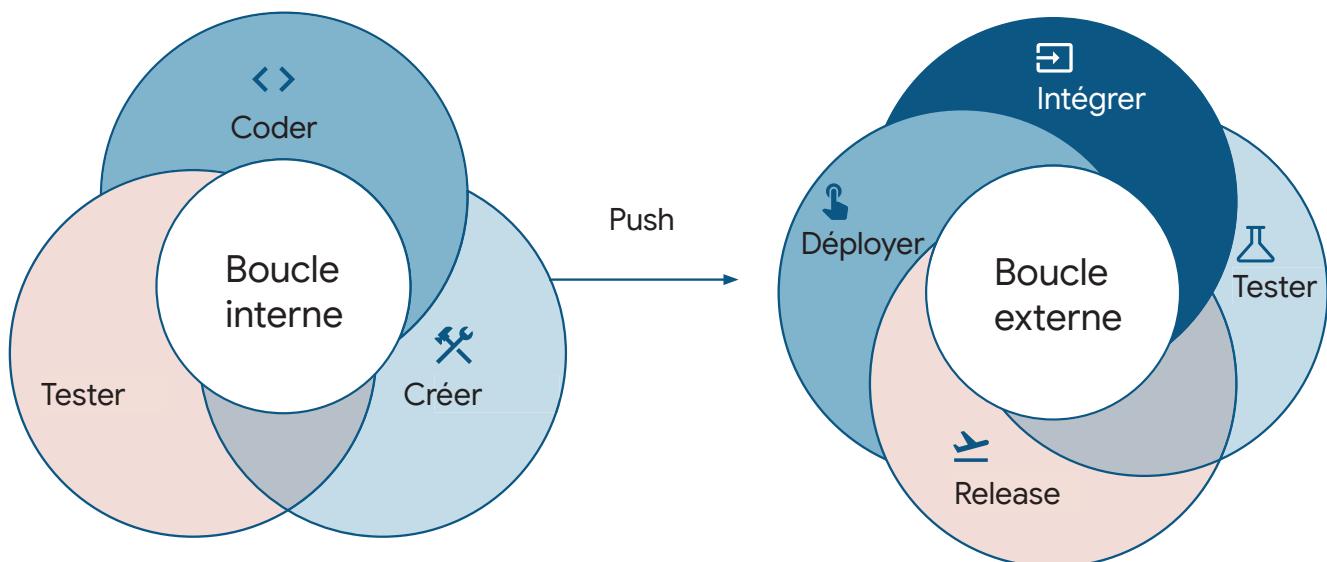
La fiabilité est une entreprise humaine, et à bien des égards, l'approche en ingénierie SRE en est un exemple. L'un des principes fondamentaux de l'ingénierie SRE consiste à considérer la perception de l'utilisateur comme le véritable indicateur de fiabilité, plutôt que les données de surveillance internes. Il n'est donc peut-être pas surprenant que la fiabilité soit motivée par une dynamique d'équipe positive. Nous avons constaté que les équipes ayant adopté une culture "générative" (qui met l'accent sur la confiance et sur la coopération) sont plus à même d'adopter des pratiques d'ingénierie SRE, et d'obtenir de bons résultats de fiabilité. Des équipes stables, dont la composition est constante dans le temps, participent également d'une plus grande fiabilité pour les services destinés aux utilisateurs. De plus, l'amélioration des actions humaines via des processus et des outils contribue aux efforts d'ingénierie en fiabilité, tout comme le DevOps en général. Des pratiques telles que l'utilisation du cloud computing et l'intégration continue sont des indicateurs de meilleurs résultats de fiabilité.

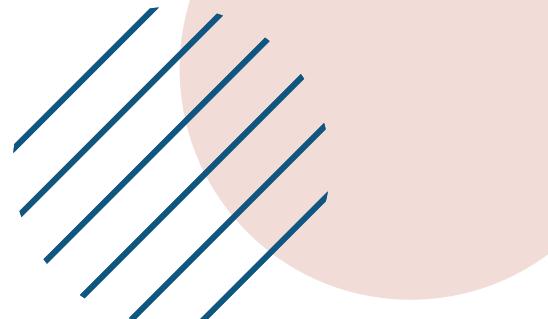


Eric Maxwell

## Capacités DevOps techniques

Cette année, nous avons examiné diverses capacités techniques pour analyser les résultats générés par différentes pratiques techniques. Nous avons observé deux grandes phases de développement logiciel : la "boucle interne", qui comprend des tâches des développeurs (codage, tests, push sur l'outil de contrôle des versions, etc.) et la "boucle externe", qui comprend les autres activités (fusion de code, revue de code automatisée, exécution de test, déploiement et release).





**Les entreprises les plus performantes qui atteignent les objectifs de fiabilité sont 1,4 fois plus susceptibles d'utiliser l'intégration continue.**

Selon nos recherches, les entreprises qui excellent dans le développement en boucles interne et externe sont capables d'expédier du code plus rapidement et avec des niveaux de fiabilité plus élevés. Les capacités qui contribuent le plus aux hautes performances sont le contrôle des versions, l'intégration continue, la livraison continue et l'architecture faiblement couplée.

**Les entreprises les plus performantes qui atteignent les objectifs de fiabilité sont :**

**33 %** plus susceptibles d'utiliser le contrôle des versions

**39 %** plus susceptibles de pratiquer l'intégration continue

**46 %** plus susceptibles de pratiquer la livraison continue

**40 %** plus susceptibles de disposer de systèmes basés sur une architecture faiblement couplée

En fait, les personnes interrogées qui utilisent plus que la moyenne l'ensemble des moyens techniques ci-dessus font état de performances organisationnelles **3,8 fois plus élevées** que celles qui s'en passent.

## Intégration continue

L'intégration continue, ou CI (Continuous Integration), est une partie du processus de développement en boucle externe qui crée automatiquement un artefact et exécute une série de tests automatisés à chaque validation de code afin d'évaluer si ce code est prêt à être déployé. Ce processus fournit un retour rapide et automatisé au développeur, lui permettant de travailler avec un niveau de confiance plus élevé. La CI joue un rôle clé lors du passage du code de la station de travail d'un développeur à la production. Comme les années précédentes, nous observons que la CI est un moteur des performances de livraison. **Les entreprises les plus performantes qui atteignent les objectifs de fiabilité sont 1,4 fois plus susceptibles d'utiliser l'intégration continue.**

Cette année, nous avons approfondi notre examen de l'autre partie du processus de développement en boucle externe : la livraison continue. Nous la décrirons dans un prochain chapitre. Mais intéressons-nous d'abord à un composant complémentaire de l'intégration continue : le développement à branche unique.

## Développement à branche unique

Le développement à branche unique consiste à fusionner le code en continu dans la branche principale et à éviter les branches de fonctionnalités de longue durée. Cette pratique est considérée comme complémentaire à l'intégration continue et accélératrice de la livraison des logiciels.

En raison de l'évolution des données démographiques en 2022 concernant les années d'expérience professionnelle des participants, nous avons pu constater l'importance de l'expérience lors de la mise en œuvre d'un processus de développement à branche unique. L'année dernière, 40 % des personnes interrogées ont déclaré qu'elles avaient plus de 16 ans d'expérience professionnelle. En 2022, cette catégorie ne représente plus que 13 %. Pour rebondir sur le thème de la dépendance de la livraison à d'autres facteurs, nous constatons que les équipes globalement moins expérimentées obtiennent de moins bons résultats avec le développement à branche unique :

- ▼ **Baisse** des performances de livraison de logiciels globales
- ▲ **Hausse** du volume de tâches imprévues
- ▲ **Hausse** de la tendance à l'erreur
- ▲ **Hausse** du taux d'échecs sur l'implémentation de modifications

Les professionnels qui ont plus de 16 ans d'expérience dans l'utilisation du développement à branche unique mesurent les avantages de cette pratique :

- ▲ **Hausse** des performances de livraison de logiciels globales
- ▼ **Baisse** du volume de tâches imprévues
- ▼ **Baisse** de la tendance à l'erreur
- ▼ **Baisse** du taux d'échecs sur l'implémentation de modifications

Cela est probablement dû aux pratiques supplémentaires requises pour réussir la mise en œuvre du développement à branche unique. Les équipes n'ayant jamais appliqué de règles rigoureuses (ex. : ne jamais ignorer une branche défectueuse) ou qui se passent de branches de code fermées et du rollback automatique du code défectueux fusionné sur la branche principale rencontreront sans doute beaucoup de difficultés avec le développement à branche unique.

Toutefois, l'utilisation du développement à branche unique a un impact positif sur les performances organisationnelles globales.



Frank Xu

## Livraison continue

Voici les avantages de la livraison continue (CD, Continuous Delivery) :

1. Elle permet aux équipes de déployer à tout moment un logiciel en production ou à destination d'utilisateurs finaux.
2. Elle garantit la déployabilité du logiciel tout au long de son cycle de vie, y compris lors du développement de nouvelles fonctionnalités.
3. Elle établit une boucle de rétroaction rapide qui permet à l'équipe de vérifier la qualité et la déployabilité du système, et donne la priorité à la résolution des problèmes bloquant le déploiement.

Veuillez noter que la livraison continue peut s'effectuer indépendamment du déploiement continu, la pratique qui consiste à déployer chaque version de logiciel automatiquement. La livraison continue exige uniquement qu'une version logicielle puisse être déployée à tout moment.

L'année dernière, nous avons étudié les capacités DevOps techniques qui indiquent qu'une équipe utilise la livraison continue. Nous avons observé que des facteurs tels qu'une architecture faiblement couplée et le test/l'intégration continu figurent parmi les indicateurs les plus fiables. Cette année, nous avons non seulement étudié les facteurs à l'origine de l'utilisation de la livraison continue, mais aussi analysé et identifié les effets de la CD elle-même ainsi que ses liens avec d'autres capacités DevOps sur le développement.

### La CD stimule les performances de livraison de logiciels

Comme les années précédentes, l'utilisation de la CD (à la fois seule et en association avec d'autres capacités DevOps) est un indicateur de performances de livraison de logiciels plus élevées. Les équipes qui obtiennent une note plus élevée en termes de CD sont plus susceptibles de déployer plus fréquemment du code en production et de raccourcir les délais pour implémenter des modifications et restaurer le service.

**Les équipes qui utilisent à la fois le contrôle des versions et la livraison continue ont 2,5 fois plus de chances de présenter des performances de livraison de logiciels élevées que les équipes qui n'utilisent qu'une seule de ces capacités.**

De plus, les personnes interrogées sont 2,5 fois plus susceptibles de signaler des performances de livraison de logiciels plus élevées lorsqu'elles ont adopté le contrôle des versions en parallèle.

### **La CD peut augmenter le volume de tâches imprévues**

Les données suggèrent que la livraison continue conduit les développeurs à consacrer davantage de temps aux corrections ou aux tâches imprévues. Les développeurs sont peut-être plus susceptibles de créer des applications de manière itérative lorsque les boucles de rétroaction sont plus étroites. Par conséquent, il est possible qu'ils considèrent certains changements itératifs comme des tâches non planifiées sur la même partie du système. Ces tâches peuvent être à la fois imprévues et motivées par les retours concernant un déploiement précédent.

### **Pratiques techniques et CD**

Nos recherches indiquent régulièrement qu'un large éventail de capacités techniques permet de mettre en œuvre la CD. Cette année, nous avons examiné ce qui se passe lorsque certaines de ces capacités individuelles sont utilisées conjointement avec la CD. Nous avons constaté que le développement à branche unique et l'architecture faiblement couplée peuvent avoir un impact négatif sur les performances d'une équipe lorsqu'ils sont utilisés en association avec la CD. Par exemple, selon notre étude, les équipes qui adoptent à la fois des architectures faiblement couplées et la CD sont 43 % plus susceptibles d'anticiper une tendance à l'erreur supérieure à la moyenne (indisponibilités du produit, failles de sécurité, dégradation importante des performances pouvant impacter leurs services) par rapport aux équipes qui n'ont adopté que la CD. Ces effets nécessitent un examen plus approfondi et suggèrent l'existence de frictions potentielles pour les équipes en progression. Ces frictions peuvent être liées à la courbe en J de la transformation, selon laquelle les équipes rencontrent d'abord des améliorations rapides, puis échouent à mesure qu'elles se penchent sur des objectifs moins accessibles. Les équipes doivent impérativement s'engager en faveur de l'amélioration pour obtenir des performances optimales. Lorsque vous améliorez une capacité, telle que la CD, vous devez surveiller l'impact des changements sur l'équipe et sur les performances globales.



David Farley

## Architecture faiblement couplée

Les systèmes faiblement couplés sont importants pour l'efficacité des équipes et des organisations. Cela ne s'applique pas seulement au cloud ou aux systèmes basés sur les microservices, mais concerne également la capacité d'une organisation à mettre en œuvre des changements. La capacité d'une organisation à modifier son logiciel avec confiance et en toute sécurité est un bon indicateur de la qualité logicielle.

Les architectures faiblement couplées offrent plusieurs possibilités pour les équipes :

- Modifier à grande échelle la conception de leur système sans dépendre de modifications apportées par d'autres équipes à leurs propres systèmes
- Obtenir des retours plus rapidement grâce à des tests indépendants et à la demande, avec des coûts de coordination réduits
- Déployer du code avec un temps d'arrêt négligeable

Dans le rapport 2022, nous avons demandé aux personnes interrogées de définir si oui ou non le logiciel qu'elles développent est basé sur une architecture faiblement couplée. Les résultats intrigants ont montré une variété d'associations, pour la plupart positives, entre l'existence d'une architecture faiblement couplée et les performances des équipes selon plusieurs dimensions.



## **Les avantages d'une architecture faiblement couplée**

Les équipes qui se focalisent sur la création de logiciels avec des architectures faiblement couplées sont mieux placées pour obtenir de solides performances en termes de stabilité, de fiabilité et de productivité. Ces équipes sont également plus susceptibles de recommander leur lieu de travail à un ami ou à un collègue.

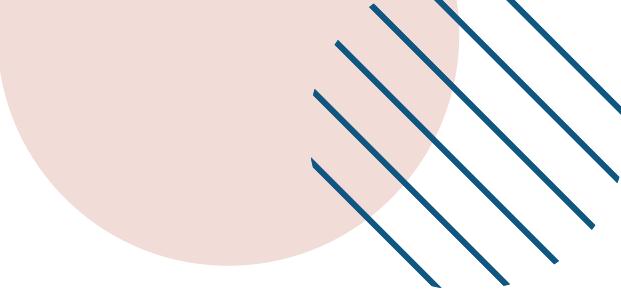
Les équipes qui déploient dans le cloud et qui adoptent une approche architecturale de microservices, assurant la gestion de centaines de services,

**Les équipes qui se focalisent sur la création de logiciels avec des architectures faiblement couplées sont mieux placées pour obtenir de solides performances en termes de stabilité, de fiabilité et de productivité.**

conçoivent souvent des logiciels basés sur une architecture faiblement couplée. Cependant, le couplage faible est plus qu'une simple mesure du nombre de services au sein d'un système. Les composants d'une architecture faiblement couplée peuvent être déployés indépendamment. Cela permet aux équipes de développer, tester et déployer leurs services de leur côté en évitant des frais de coordination importants.

En réalité, le couplage faible ne se limite pas à un seul style d'architecture. Fondamentalement, il s'agit de la capacité d'effectuer un changement dans une partie du système, sans que ce changement ait un impact sur les autres parties. Cela permet aux organisations de diviser leurs tâches, de sorte que les équipes individuelles puissent progresser sans avoir à se coordonner avec d'autres équipes.

Selon notre expérience, les équipes qui effectuent des tests d'intégration approfondis avec d'autres services pour renforcer la confiance dans leur logiciel avant son déploiement ne sont pas encore véritablement passées au couplage faible. Pour atteindre cet objectif, elles gagneraient à optimiser les interfaces et l'isolement des systèmes, par exemple en améliorant la "testabilité" des services et des composants. Si votre infrastructure vous permet de tester votre service de manière isolée, son interface est, par définition, faiblement couplée.



Nous avons également constaté que les équipes solides et stables qui utilisent une architecture faiblement couplée sont plus susceptibles de mettre en œuvre des pratiques de développement logiciel qui encouragent et stimulent l'amélioration continue. Par exemple, certaines pratiques d'ingénierie SRE contribuent à une architecture faiblement couplée (comme définir des objectifs de fiabilité pour hiérarchiser les tâches, ou encore réaliser des examens réguliers visant à réviser les objectifs de fiabilité en fonction de faits avérés).

Les architectures faiblement couplées permettent également à l'organisation d'employer davantage de personnel : les équipes, autonomes et non tenues de se coordonner, sont plus libres de croître en toute indépendance.

En bref, l'impact du couplage faible des services logiciels ne se résume pas aux aspects strictement techniques. La dimension sociotechnique du développement logiciel est également concernée. Le couplage est un concept fondamental de la loi de Conway qui signifie que les systèmes de conception d'une organisation reflètent leur propre structure de communication. Qui dit systèmes plus faiblement couplés, dit organisations plus faiblement couplées avec une approche du développement plus distribuée et évolutive.



## Des résultats surprenants

Les recherches menées cette année ont révélé que l'architecture faiblement couplée pourrait contribuer au burnout des équipes. Il s'agit d'une découverte surprenante qui contredit les résultats des années précédentes. Selon notre analyse, les équipes stables dans lesquelles l'information circule librement sont moins sujettes au burnout. La culture générative de Westrum et la stabilité des équipes font partie des fondements d'une architecture faiblement couplée et contribuent à la réduction du burnout, ce qui est donc clairement contradictoire. Des recherches complémentaires sont nécessaires avant de pouvoir tirer des conclusions définitives.

Dans le même temps, lorsque les exigences de sécurité sont définies et contrôlées par une organisation de sécurité consolidée, il peut être plus difficile pour les équipes de dissocier leur logiciel des autres équipes. Il est donc bien avantageux de transmettre les problèmes de sécurité à l'équipe la plus impliquée dans le développement de l'application (voir également : [L'importance de la sécurité sur la chaîne d'approvisionnement](#)). Cet aspect constitue l'une des formes les plus subtiles de couplage dans les organisations, et même si les données que nous avons recueillies concernent la sécurité, il est probable que les autres fonctions centrales soient également concernées. Autoriser les équipes à prendre leurs propres décisions en termes de sécurité et d'autres fonctions centrales est un moyen pour votre organisation de récolter progressivement les bénéfices des architectures faiblement couplées.



Daniella Villalba

## Culture

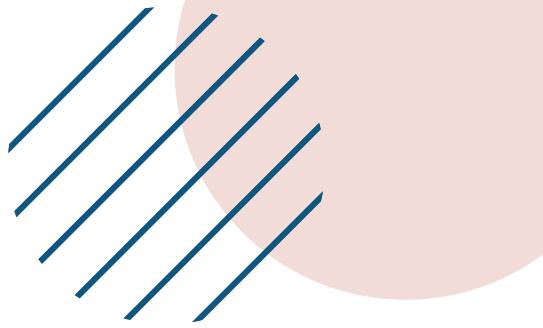
"Ici, c'est comme ça que ça marche."

Quel que soit le secteur d'activité, tout le monde a probablement prononcé cette phrase au moins une fois pour décrire l'approche de son organisation face aux défis et aux opportunités.

Au fil du temps, nos recherches ont montré que la culture, propre à chaque organisation, est essentielle à sa réussite et au bien-être de ses employés.

La culture constitue également un aspect nécessaire du DevOps. En effet, au niveau le plus élémentaire, le DevOps concerne les outils, les pratiques et **les méthodes de coopération entre les personnes** pour développer et fournir des logiciels rapidement, de manière fiable et en toute sécurité. L'analyse des facteurs qui ont un impact sur la culture d'une organisation peut aider sa direction à relever les défis culturels. Par conséquent, instaurer une culture saine devrait être une priorité. S'ils ne sont pas résolus, les défis culturels peuvent empêcher la mise en place des pratiques DevOps.





Cette année, nous nous sommes à nouveau appuyés sur la typologie organisationnelle de Westrum pour mesurer la santé de la culture d'une organisation. En outre, nous avons élargi notre analyse culturelle en mesurant le renouvellement des équipes, la flexibilité des conditions de travail, l'adhésion organisationnelle perçue et le burnout.

Les données de l'étude de 2022 confirment les conclusions précédentes selon lesquelles les **performances organisationnelles** sont influencées par le type de culture en place au sein d'une organisation. Plus précisément, une **culture génératrice** est associée à des performances organisationnelles plus élevées par rapport aux organisations de type bureaucratique ou

pathologique. Dans les organisations à culture génératrice, les employés sont plus à même d'appartenir à des équipes stables, de produire une documentation de meilleure qualité et de se consacrer à des tâches utiles.

## Renouvellement des équipes

Selon nos recherches, les **équipes stables** (dont la composition n'a pas beaucoup changé en 12 mois) sont davantage présentes dans les organisations aux performances élevées. Le renouvellement constant des équipes peut impacter la productivité et le moral du groupe, car l'accueil des nouveaux membres prend du temps. Ceux qui restent

## Culture organisationnelle Westrum

Pathologique	Bureaucratique	Générative
Orientée vers le pouvoir	Orientée vers les règles	Orientée vers la performance
Faible coopération	Coopération modeste	Grande coopération
Messagers "abattus"	Messagers négligés	Messagers formés
Responsabilités non assumées	Responsabilités limitées	Responsabilités partagées
Communication découragée	Communication tolérée	Communication encouragée
L'échec conduit à se rejeter la faute	L'échec conduit au tribunal	L'échec conduit à une enquête
La nouveauté est écrasée	La nouveauté crée des problèmes	La nouveauté est mise en œuvre



doivent parfois s'adapter à l'évolution de leur charge de travail et à une nouvelle dynamique d'équipe. Par ailleurs, nos recherches ont montré que les équipes stables étaient plus susceptibles de déclarer produire une documentation de qualité que les équipes qui changent plus souvent. Une équipe qui doit constamment faire face au changement peut avoir plus de mal à conserver des habitudes permettant de produire une documentation de qualité.

**Les organisations aux performances élevées sont plus susceptibles de proposer des conditions de travail flexibles.**

## Flexibilité des conditions de travail

Au regard de la transition vers des conditions de travail flexibles que de nombreuses organisations ont adoptées depuis le début de la pandémie de COVID-19, nous avons cherché à savoir si un lien existe entre le choix du lieu de travail (télétravail, bureau, les deux) et des performances organisationnelles plus élevées.

Nos résultats indiquent que les organisations offrant un niveau plus élevé de **flexibilité à leurs employés** gagnent en performances organisationnelles par rapport aux organisations dont les conditions de travail sont plus rigides. Ces résultats prouvent que le fait de laisser les employés choisir leur lieu de travail bénéficie de manière directe et tangible aux organisations.

## Burnout

Le burnout est un sentiment de terreur, d'apathie et de cynisme entourant le travail. Lorsque les employés sont victimes du burnout, ils sont non seulement démotivés et épuisés, mais aussi généralement moins satisfaits de leur emploi, ce qui peut augmenter la fréquence de renouvellement du personnel. Le burnout est associé à un large éventail de problèmes de santé psychologique et physique, tels qu'un risque accru de dépression et d'anxiété, de maladies cardiaques et de pensées suicidaires<sup>1</sup>.

L'année dernière, nous avons mesuré l'épuisement professionnel dans le contexte de la pandémie de COVID-19 et constaté qu'une culture génératrice était associée à des taux plus faibles de burnout des employés.

<sup>1</sup> Maslach C, Leiter MP. Understanding the burnout experience: recent research and its implications for psychiatry. World Psychiatry. 2016 Jun;15(2):103-11. doi: 10.1002/wps.20311. PMID: 27265691; PMCID: PMC4911781.

**Les modèles de travail flexibles sont associés à une diminution du burnout des employés et à une augmentation de la probabilité qu'ils recommandent leur équipe en tant qu'environnement de travail idéal.**

Cette année, nous avons confirmé ce résultat et élargi notre analyse du burnout en démontrant que la stabilité des équipes et la flexibilité des conditions de travail font également baisser le burnout. De plus, cette année, nous avons mesuré le Net Promoter Score (NPS) des équipes, qui indique la probabilité que les membres d'une équipe la recommandent à un ami ou à un collègue. Nos résultats suggèrent qu'un lien existe entre le NPS d'une équipe et le sentiment d'être soutenu par la direction. Et tout comme pour le burnout, nous avons constaté que l'association d'une culture génératrice, d'une équipe stable et de conditions de travail flexibles avait une influence sur la propension des personnes à recommander leur équipe à des tiers.

## Comment les employés perçoivent leur organisation

Enfin, nous avons cherché à savoir dans quelle mesure les personnes interrogées se sentaient soutenues par leur direction. Pour ce faire, nous leur avons demandé d'estimer le niveau de soutien qu'elles s'attendaient à recevoir en tant qu'équipe au cours des 12 mois suivants. Les résultats indiquent que lorsque les équipes se sentent davantage soutenues par leur direction (ex. : budget plus élevé, ressources plus importantes, parrainages), les organisations sont plus performantes.

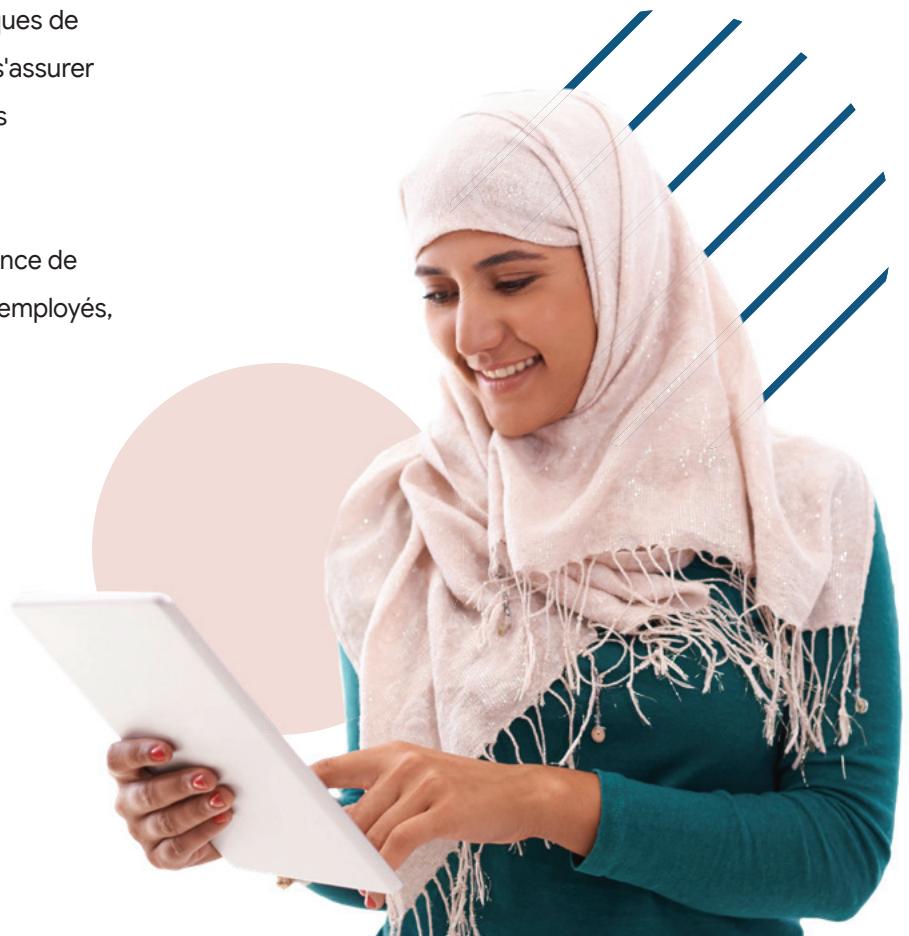
Nous avons également demandé aux personnes interrogées d'estimer la probabilité qu'une brèche de sécurité ou une indisponibilité complète se produise au cours des 12 mois suivants. Les résultats indiquent que les employés des organisations très performantes sont moins susceptibles de craindre qu'un problème majeur se produise, car ils ont une vision plus positive de leur organisation. De même, les employés des organisations ayant des performances logicielles et de livraison élevées sont **moins susceptibles** de penser que leurs pratiques actuelles doivent évoluer pour de meilleurs résultats commerciaux.

## Quelques mots sur la diversité

Selon nos résultats, les employés appartenant à des groupes sous-représentés sont plus susceptibles de déclarer consacrer davantage de temps à des tâches imprévues, quel que soit le niveau de performance de leur organisation. Par ailleurs, les employés appartenant aux groupes sous-représentés signalent des niveaux de burnout plus élevés que les autres employés. Les chefs d'équipe doivent être conscients des risques de déséquilibre en termes de charge de travail et s'assurer que le travail est réparti équitablement entre les membres de l'équipe.

Pris ensemble, ces résultats soulignent l'importance de créer un environnement sain et inclusif pour les employés, tant au niveau de l'organisation que de l'équipe.

Bien que nous continuions à souligner l'importance de la culture interne, nous reconnaissons que changer ou même améliorer la culture d'une organisation n'est pas une tâche facile. Nous recommandons aux organisations de chercher avant tout à comprendre les épreuves que rencontrent leurs employés, puis d'investir des ressources dans la résolution des problèmes liés à la culture dans le cadre des efforts de transformation DevOps.



# 04

## L'importance de la sécurité sur la chaîne d'approvisionnement

En novembre 2020, relativement peu de professionnels de la technologie soupçonnaient qu'une crise de la sécurité sur la chaîne d'approvisionnement des logiciels se préparait. L'[Open Source Security Foundation](#) avait été formée en réponse aux efforts passés pour veiller à la sécurité des logiciels Open Source. Et si quelques [initiatives notables](#) avaient vu le jour pour faire face à ce problème, le sujet intéressait peu. C'est une attaque majeure, [SolarWinds](#), qui a tout changé. Lorsque des pirates informatiques ont espionné sans se

faire repérer des milliers de grandes entreprises et de réseaux gouvernementaux grâce à des mises à jour de logiciels contenant un cheval de Troie, les réactions ne se sont pas fait attendre.

Aujourd'hui, le sujet de la sécurité sur la chaîne d'approvisionnement logicielle est devenu largement prioritaire, tout du moins dans un contexte professionnel.



John Speed Meyers Todd Kulesza



De nombreuses initiatives ont été lancées, et une grande partie de l'industrie logicielle s'est engagée non seulement à réformer ses propres pratiques de sécurité sur la chaîne d'approvisionnement logiciel, mais aussi à améliorer la sécurité des ressources Open Source communes.

Dans ce chapitre, nous abordons deux de ces initiatives : Supply-chain Levels for Software Artifacts ([SLSA, se prononce "salsa"](#)) ou "Niveaux de la chaîne d'approvisionnement pour les artefacts logiciels", et NIST Secure Software Development Framework ([SSDF](#)) ou "Framework de développement de logiciels sécurisés du NIST". Chaque initiative propose un ensemble de mesures défensives pour s'assurer que les pirates informatiques ne peuvent pas altérer les processus de production de logiciels et passer outre les défenses du réseau via des mises à jour malveillantes de logiciels.

Mais dans quelle mesure les entreprises s'appuient-elles sur des pratiques de sécurité sur la chaîne d'approvisionnement logiciel liées à SLSA et SSDF ? Quelles sont les pratiques les plus difficilement adoptées et, au contraire, les plus répandues ? À ce jour, nous n'avons pas de réponses systématiques à ces questions, mais l'enquête menée auprès de centaines de professionnels des logiciels concernant leurs pratiques en termes de sécurité sur la chaîne d'approvisionnement fait ressortir quatre pistes principales :

**01 Le processus d'adoption a commencé :** Les pratiques SLSA et SSDF de sécurité sur la chaîne d'approvisionnement logiciel sont déjà modestement adoptées, mais le potentiel de progression est important.

**02 Les organisations qui favorisent une culture saine ont une longueur d'avance :** La culture organisationnelle est l'un des principaux moteurs des pratiques de sécurité du développement de logiciels. Les organisations qui évitent de développer une culture de reproche ont un niveau de confiance mutuelle plus élevé et sont plus susceptibles d'établir des pratiques SLSA et SSDF que les organisations présentant un niveau de confiance mutuelle faible.

**03 Un point d'intégration central existe :** L'adoption des aspects techniques de la sécurité logicielle sur la chaîne d'approvisionnement semble dépendre de l'utilisation du binôme CI/CD, qui fournit souvent la plate-forme d'intégration de nombreuses pratiques de sécurité sur la chaîne d'approvisionnement.

**04 L'adoption de bonnes pratiques offre des avantages inattendus :** Outre une réduction des risques de sécurité, des avantages supplémentaires apparaissent, comme la réduction du burnout.

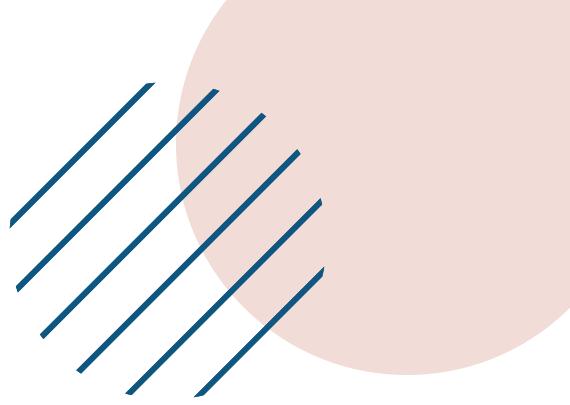
## Mesures actuelles prises par les entreprises pour éviter les failles de sécurité

Pour mieux comprendre comment les organisations s'organisent de nos jours pour identifier et résoudre les failles de sécurité dans les logiciels qu'elles conçoivent, nous avons ajouté plus d'une vingtaine de questions à l'enquête 2022. Ces questions se divisent globalement en deux catégories :

- Questions de type "D'accord/Pas d'accord avec une affirmation" (par exemple, "Mon organisation dispose d'une solution efficace pour combattre les menaces de sécurité" ou "J'ai accès à l'éventail d'outils nécessaires pour procéder à des tests de sécurité").
- Questions cherchant à mesurer l'état de mise en œuvre des pratiques de sécurité dans l'organisation (par exemple, "Les builds sont uniquement définis via le script de compilation" ou "Les versions en production sont compilées à l'aide d'un système CI/CD centralisé et en aucun cas sur la station de travail d'un développeur"). Nous avons utilisé ce type d'échelle de réponse, car nos premiers tests ont révélé que les personnes interrogées avaient tendance à orienter leurs réponses en faveur des affirmations concernant la sécurité. Les questions sur le framework SSDF, en revanche, étaient formulées de façon plus naturelle avec l'échelle de réponse "D'accord/Pas d'accord".

Le **framework SLSA** (dans sa version 0.1 au moment de la rédaction du rapport) expose une série de pratiques d'intégrité sur la chaîne d'approvisionnement logicielle associées aux "niveaux" SLSA, les niveaux supérieurs correspondant à une meilleure confiance dans la sécurité sur la chaîne d'approvisionnement logicielle. Nous avons interrogé les personnes sur bon nombre de pratiques particulières associées au framework SLSA. Plus précisément, la question suivante était posée : "Dans quelle mesure les pratiques suivantes sont-elles établies pour l'application ou le service que vous développez en priorité ?" Le tableau 1 recense les intitulés des pratiques liées au framework SLSA abordées dans l'enquête.

Le **framework SSDF** (actuellement dans sa version 1.1) se concentre sur les pratiques qui permettent aux organisations de livrer des logiciels moins sujets aux failles et dont l'impact des failles existantes est moindre. Les pratiques SSDF ne sont pas regroupées par niveaux comme dans le framework SLSA, mais sont divisées en quatre catégories : préparation de l'organisation, protection des logiciels en cours de développement, conception de logiciels correctement sécurisés et réponse efficace aux failles découvertes. Les personnes interrogées devaient déclarer si elles étaient d'accord ou pas d'accord avec des affirmations relatives à plusieurs pratiques du SSDF (voir le tableau 2 pour un résumé de ces questions).



Pratique SLSA	Définition de l'enquête
<b>CI/CD centralisée</b>	Les versions en production sont compilées à l'aide d'un système CI/CD centralisé et en aucun cas sur la station de travail d'un développeur
<b>Conservation de l'historique</b>	Les révisions et l'historique des modifications sont conservés indéfiniment
<b>Script de compilation</b>	Les builds sont intégralement et uniquement définis via le script de compilation
<b>Isolement</b>	Les builds sont isolés et ne peuvent pas interférer avec les builds simultanés ou ultérieurs
<b>Fichier texte de build</b>	Les définitions et les configurations des builds sont définies dans des fichiers texte stockés dans un système de contrôle des versions
<b>Métadonnées de paramètres</b>	Les métadonnées d'artefact de build (dépendances, processus et environnement de compilation, etc.) comprennent tous les paramètres de compilation
<b>Métadonnées de dépendances</b>	Les métadonnées d'artefact de build (dépendances, processus et environnement de compilation, etc.) listent toutes les dépendances
<b>Génération de métadonnées</b>	Les métadonnées de build (dépendances, processus et environnement de compilation, etc.) sont générées par le service de compilation ou par un générateur de métadonnées de build capable de lire le service de compilation
<b>Blocage des entrées</b>	Lors des différentes étapes d'exécution de la compilation, aucune entrée de build ne peut être chargée dynamiquement (en d'autres termes, toutes les sources et dépendances requises sont extraites à l'avance)
<b>Modification par l'utilisateur interdite</b>	Les métadonnées d'artefact de build (dépendances, processus et environnement de compilation, etc.) ne peuvent pas être modifiées par les utilisateurs des services de compilation
<b>Disponibilité des métadonnées</b>	Les métadonnées de build (dépendances, processus et environnement de compilation, etc.) sont disponibles pour quiconque en a besoin (via une base de données centrale, par exemple) et sont accessibles dans un format compatible
<b>Examen en binôme</b>	Chaque modification de l'historique d'une révision doit être examinée et approuvée individuellement par deux personnes de confiance avant l'envoi
<b>Signature des métadonnées</b>	Les métadonnées sur la production d'un artefact de build (dépendances, processus et environnement de compilation, etc.) sont signées par le service de compilation

**Tableau 1. Questions de l'enquête liées au SLSA**

Remarque : Cinq options de réponse étaient proposées aux participants : pas du tout établi, légèrement établi, moyennement établi, très établi et complètement établi.

Pratique SSDF	Définition de l'enquête
<b>Examens de sécurité</b>	Un examen de sécurité est effectué pour toutes les fonctionnalités majeures des applications que je développe
<b>Analyse et test continu du code</b>	Des analyses et des tests de code automatisés ou manuels sont mis en œuvre continuellement pour toutes les versions disponibles, afin d'identifier ou de confirmer la présence de failles non détectées auparavant
<b>Tests de sécurité précoce</b>	Des tests de sécurité sont exécutés au début du processus de développement logiciel par un responsable (moi-même ou une autre équipe)
<b>Traitement efficace des menaces</b>	Mon organisation dispose d'une solution efficace pour combattre les menaces de sécurité
<b>Intégration à l'équipe de développement</b>	Les rôles responsables de la sécurité sont intégrés à l'équipe de développement logiciel
<b>Exigences de documentation</b>	Mon organisation a mis en place des processus permettant d'identifier et de documenter toutes les exigences de sécurité pour les logiciels qu'elle développe ou qu'elle possède (y compris les logiciels tiers et Open Source)
<b>Examen régulier des exigences</b>	Les exigences de sécurité sont examinées à intervalles réguliers (une fois par an ou plus si nécessaire)
<b>Génération de métadonnées</b>	Les métadonnées de build (dépendances, processus et environnement de compilation, etc.) sont générées par le service de compilation ou par un générateur de métadonnées de build capable de lire le service de compilation
<b>Intégration au cycle de développement</b>	Mon entreprise dispose d'un protocole de sécurité logicielle parfaitement intégré au processus de développement
<b>Processus standard commun à tous les projets</b>	Mon entreprise a mis en place un processus standard commun à tous les projets concernant la sécurité des logiciels
<b>Surveillance des rapports de sécurité</b>	Un suivi continu des informations provenant de sources publiques nous permet d'identifier les failles potentielles dans le logiciel utilisé et ses composants tiers
<b>Disponibilité des outils nécessaires</b>	J'ai accès à l'éventail d'outils nécessaires pour procéder à des tests de sécurité

**Tableau 2. Questions de l'enquête liées au SSDF**

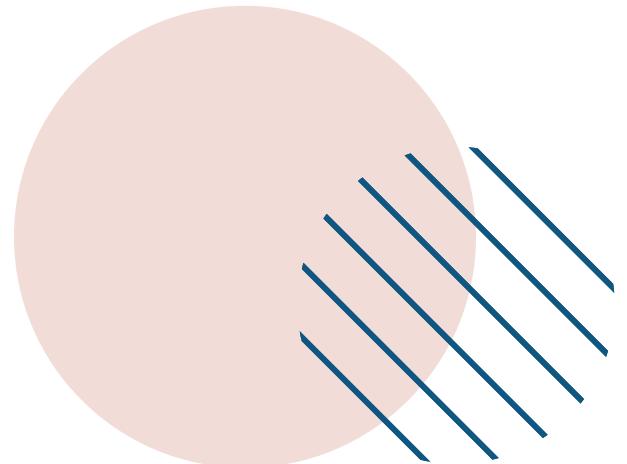
Remarque : Sept options de réponse étaient proposées aux participants : pas du tout d'accord, pas d'accord, plutôt pas d'accord, sans opinion, plutôt d'accord, d'accord et tout à fait d'accord.

Dans l'ensemble, nous avons constaté une adoption assez large des pratiques émergentes du secteur, avec toutefois une marge de manœuvre importante pour les établir davantage. Par exemple, tandis que 66 % des personnes interrogées sont d'accord avec l'énoncé "Mon entreprise dispose d'un protocole de sécurité logicielle parfaitement intégré au processus de développement", seuls 18 % sont tout à fait d'accord. Les figures 1 et 2 résument les réponses des participants à nos questions sur la sécurité.

Selon nos résultats, l'utilisation de systèmes d'intégration continue/livraison continue (CI/CD) pour les versions de production est la pratique la plus couramment établie : 63 % des personnes interrogées déclarent qu'elle est "très" ou "complètement" établie. Le fait que la CI/CD arrive en tête de cette liste est conforme aux recherches antérieures sur la sécurité, qui ont révélé [que la plupart des organisations mettent en œuvre une analyse de sécurité au niveau des applications dans le cadre de leur processus CI/CD](#). En outre, une série distincte d'entretiens qualitatifs axés sur la sécurité a révélé que la plupart des développeurs n'étaient pas en mesure d'exécuter ces outils localement au cours du développement. Le framework SLSA s'appuie également sur les systèmes CI en tant que point d'intégration central pour la sécurité sur la chaîne d'approvisionnement. Notre analyse du modèle, décrite dans la section suivante, a révélé que la

présence d'un système CI dans une organisation était un bon indicateur de la maturité de ses pratiques de sécurité. Ainsi, nous sommes convaincus que sans cet élément essentiel de l'infrastructure, il est très difficile pour une organisation d'assurer l'exécution d'un ensemble cohérent d'outils d'analyse, de lint et de test sur les artefacts logiciels qu'ils conçoivent.

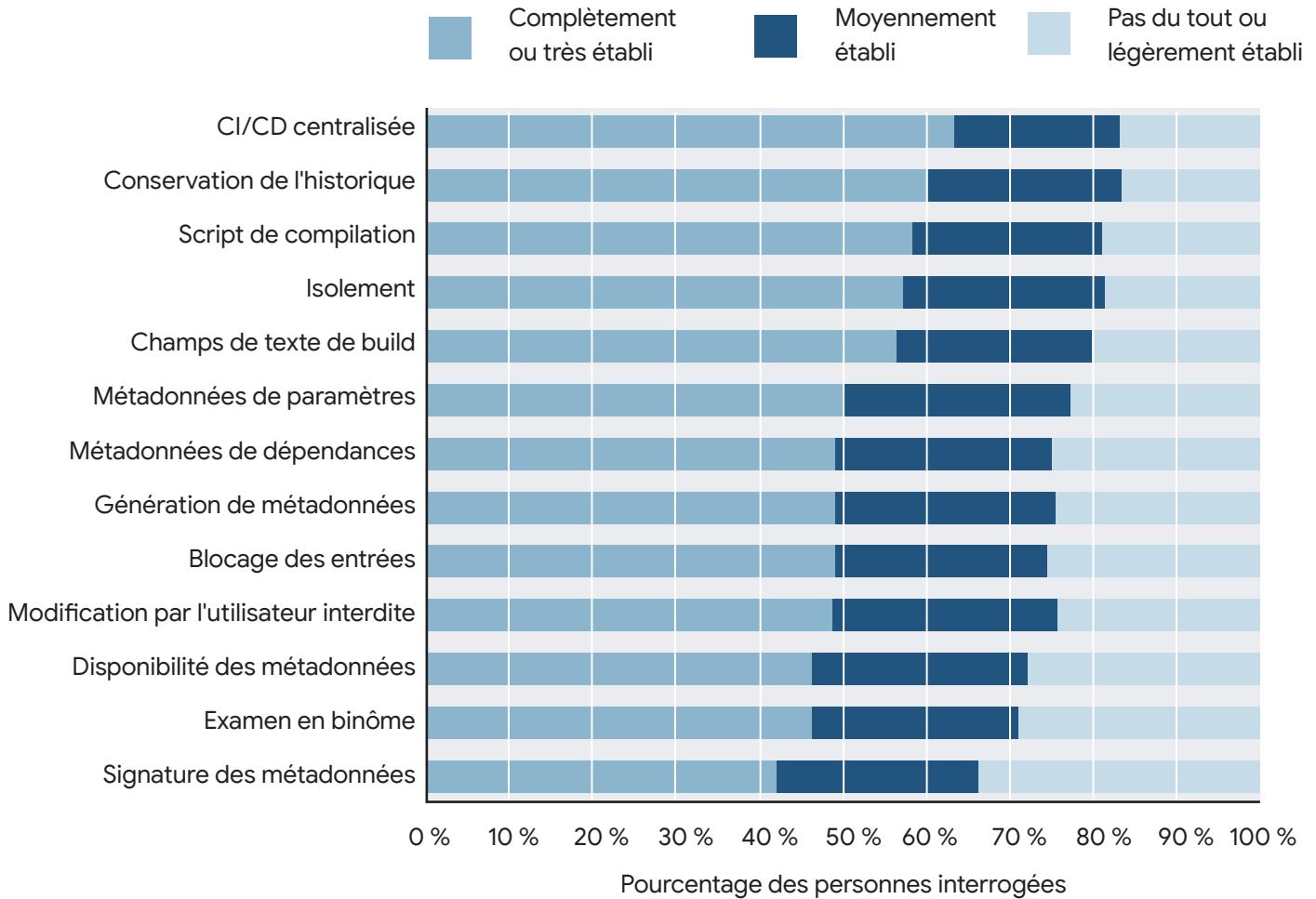
Outre la CI/CD, d'autres pratiques sont couramment établies, comme la conservation de l'historique du code sans limite de temps (60 %), la définition des builds uniquement à l'aide de scripts (58 %), l'isolement des builds entre eux (57 %) et le stockage des définitions de build dans le contrôle du code source (56 %). À l'inverse, les deux pratiques les moins couramment établies sont l'examen et la validation en binôme des modifications de code (45 %) et la signature des métadonnées de build pour empêcher/détecter les accès non autorisés (41 %).



En parallèle des questions sur le degré d'adoption des pratiques, nous avons cherché à savoir si les participants étaient d'accord ou non avec un ensemble d'énoncés sur la sécurité au sein de leur organisation. Voici l'énoncé ayant recueilli le plus d'adhésion (81 %) : "Un suivi continu des informations provenant de sources publiques nous permet d'identifier les failles potentielles dans le logiciel utilisé et ses composants tiers". À l'opposé, celui ayant recueilli le moins d'adhésion concerne l'impact négatif des pratiques de sécurité sur le développement de logiciels : 56 % des personnes interrogées sont d'accord avec l'énoncé "Les processus de sécurité logicielle à l'œuvre dans mon entreprise ralentissent le processus de développement des applications sur lesquelles je travaille."

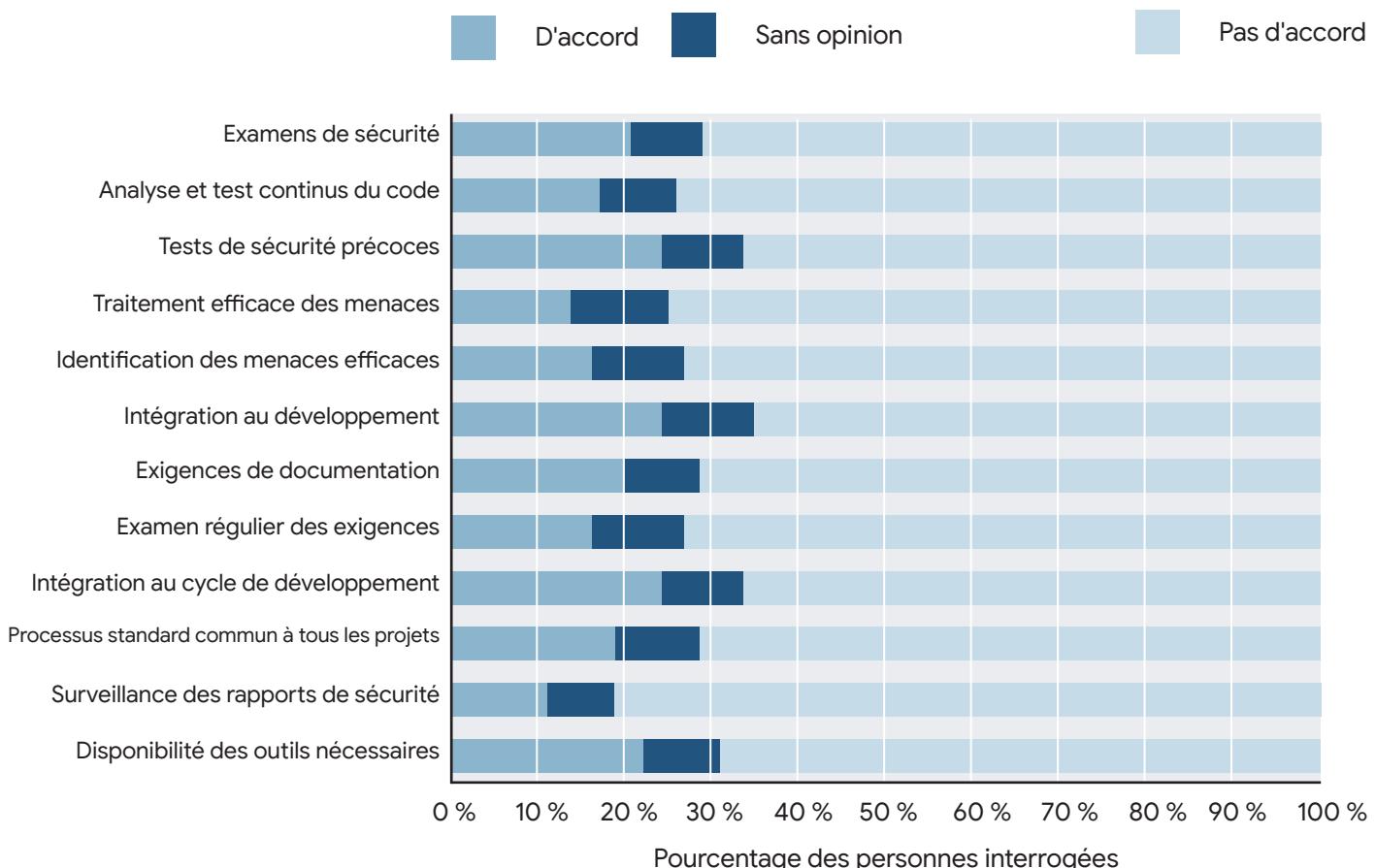
Bien que cet énoncé arrive dernier en pourcentage d'adhésion, une majorité de répondants pense que les processus de sécurité actuels ralentissent le développement, ce qui suggère qu'une grande marge d'amélioration des outils et des approches existe en termes de sécurité. L'analyse de notre modèle corrobore également cette interprétation, montrant des effets mitigés (bien que mineurs) sur les performances de livraison de logiciels.





**Figure 1. Mise en place des pratiques SLSA**

Réponses à l'enquête sur la mise en place des pratiques SLSA. Une majorité de personnes interrogées ont indiqué un certain degré de mise en place pour toutes ces pratiques, mais assez peu ont déclaré qu'elles étaient déjà "complètement" établies.



**Figure 2. Mise en place des pratiques SSDF**

Réponses à l'enquête sur la mise en place des pratiques SSDF. Comme pour la SLSA, une majorité de personnes interrogées ont reconnu que leur organisation suivait l'ensemble de ces pratiques.

# Qu'est-ce qui favorise l'adoption de bonnes pratiques de sécurité ?

La sécurité des applications ne constitue qu'un aspect du développement de logiciels, auquel doivent se consacrer les développeurs parmi de nombreuses autres missions. Les approches de sécurité qui génèrent beaucoup de friction peuvent être frustrantes pour les développeurs et s'avèrent globalement inefficaces, car les équipes cherchent à contourner les points de friction. Par exemple, lors d'une série d'entretiens de recherche avec des ingénieurs logiciels professionnels, nous avons découvert que les échanges avec les équipes de sécurité se limitent au début ou à la fin des projets, et qu'il n'est pas toujours évident d'interagir avec ces équipes. L'un des participants décrit la relation en ces termes : "Une équipe est responsable de la sécurité des applications, mais je ne lui ai jamais fait vérifier mon code... En général, je fais comme les autres ingénieurs : je l'évite".

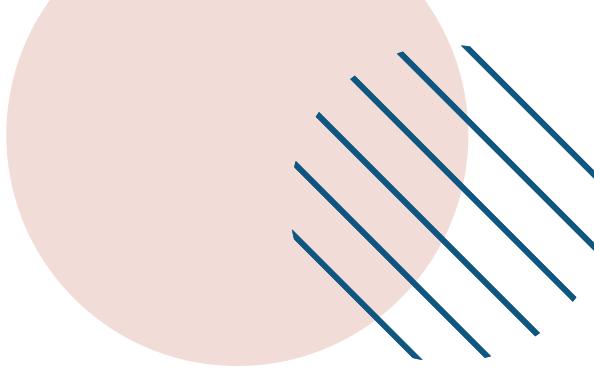
Une façon d'améliorer la sécurité des logiciels consiste à réduire les obstacles au respect des pratiques de sécurité. Les développeurs avec lesquels nous avons échangé faisaient preuve de bonne volonté et se sentaient frustrés de toujours devoir donner la priorité au lancement de fonctionnalités ou de corrections plutôt qu'aux problèmes de sécurité potentiels. Par exemple, un participant à une enquête distincte sur la sécurité décrit son plus grand défi en ces termes : "En faire une priorité par défaut. Ce n'est pas très glamour, ne contribue pas à vendre plus de produits, [et] n'est pas un problème jusqu'à ce que ça en devienne un."

Les données de notre enquête suggèrent que plusieurs facteurs permettent aux développeurs de "s'occuper de la sécurité".

Le facteur le plus important que nous avons identifié n'était pas du tout technique, mais plutôt culturel : **les organisations les plus proches du groupe culturel "génératif" selon Westrum sont beaucoup plus susceptibles d'estimer que leurs pratiques de sécurité sont bien établies** d'après le framework SLSA<sup>1</sup>. Les caractéristiques des cultures génératives incluent la capacité à coopérer à grande échelle, à partager les risques et les responsabilités et à tirer des enseignements des erreurs passées. Nous émettons l'hypothèse que ces caractéristiques se traduisent par de meilleures pratiques de sécurité de plusieurs façons, par exemple en encourageant les ingénieurs logiciels à se montrer plus proactifs en ce qui concerne la sécurité sur la chaîne d'approvisionnement, en récompensant les personnes pour leurs efforts en termes de sécurité (quel que soit leur poste) ou en réduisant l'impression que signaler des problèmes de sécurité potentiels comporte des risques.

Sur le plan technologique, trois des facteurs les plus importants pour la sécurité concernent l'infrastructure. Cela semble naturellement pertinent, car si votre infrastructure facilite la réalisation de tâches telles que l'analyse des failles ou la revue manuelle du code, il est plus probable que vos ingénieurs s'en servent. Plus précisément, **disposer de systèmes de contrôle des sources, d'intégration continue et de livraison continue permet de renforcer les pratiques SLSA.**

<sup>1</sup> Fait intéressant, ces mêmes participants n'approuvent pas davantage les énoncés du framework NIST SSDF. Même si les frameworks SLSA et SSDF abordent des aspects différents de la sécurité du développement d'applications, nous nous attendions à des similitudes entre les réponses aux deux ensembles de questions. Comme mentionné précédemment, cette différence s'explique peut-être par l'orientation de l'échelle de réponse du framework SSDF en faveur des réponses positives.



Cette association repose essentiellement sur le moment où les développeurs se rendent compte que des problèmes de sécurité existent, ce qui correspond à l'étape de l'intégration continue, selon une autre enquête. En règle générale, l'étape d'intégration continue précède directement la revue du code. C'est à cette étape que les analyses des failles et les autres outils d'analyse de code sont exécutés, car la CI garantit que tous les commits de code sont soumis aux mêmes exigences de sécurité. Sans système de compilation centralisé, l'exécution d'analyses est beaucoup plus difficile à mettre en œuvre de manière constante. De plus, le manque de contrôle des sources empêche l'adoption d'un système de compilation centralisé.

Cependant, il est possible que les ingénieurs logiciels souhaitent effectuer les analyses plus tôt qu'à l'étape de CI/CD. Dans une série d'entretiens sur le thème de la sécurité, les développeurs d'applications interrogés ont systématiquement déclaré que l'exécution des analyses de sécurité sur leur station de travail de développement pouvait leur faire gagner du temps et leur simplifier la vie. Deux situations sont couramment citées : 1) le besoin d'identifier à l'avance les failles connues dans les dépendances du développement, et ce, afin de pouvoir reconstruire la nécessité de développer du nouveau code sur ces dépendances et 2) éviter d'être confronté à de longues attentes en CI (jusqu'à plusieurs heures) pour savoir si les changements en cours ont permis de résoudre un problème de sécurité. Dans les deux cas, les ingénieurs logiciels déclarent que même si un "filet

de sécurité" CI est nécessaire, la possibilité d'exécuter les mêmes outils de sécurité en local leur permettrait de travailler plus rapidement et plus efficacement.

Les facteurs culturels et technologiques évoqués ci-dessus figurent parmi les principaux moteurs de la sécurité, mais d'autres facteurs importants existent :

- Flexibilité des conditions de travail (encourager le télétravail, par exemple)
- Utilisation du cloud (public ou privé)
- Travail sur une application ou un service "cloud natif"
- Sentiment d'être valorisé par l'entreprise et que celle-ci investit dans l'équipe
- Faible renouvellement des équipes
- Taille de l'organisation (les grandes organisations signalant des scores de sécurité plus élevés)

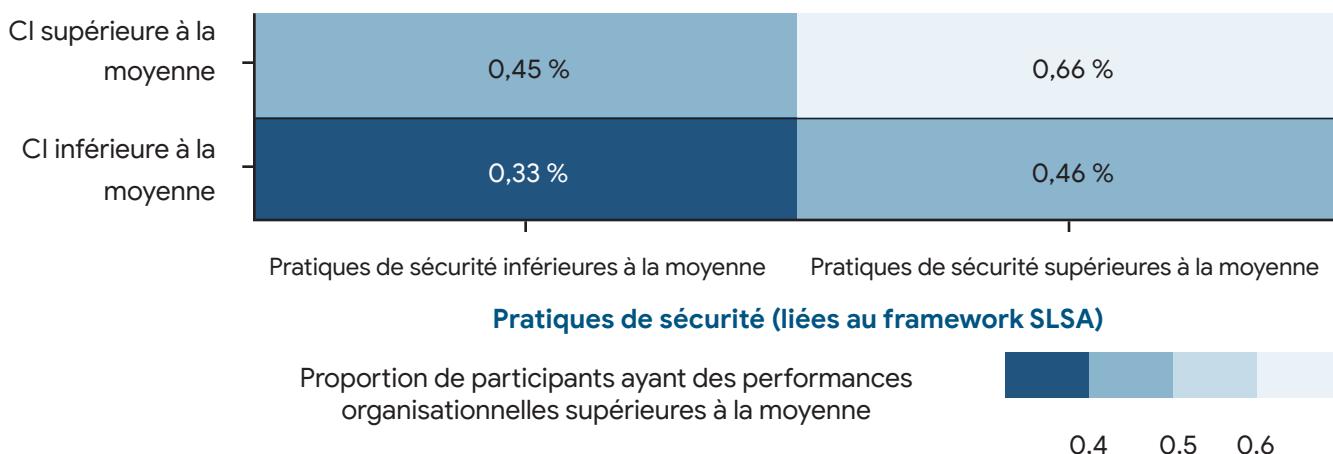
Cependant, ces facteurs semblent généralement corrélés soit à la culture génératrice de Westrum (par exemple, des conditions de travail flexibles, le sentiment d'être valorisé par l'organisation ou le faible renouvellement des équipes), soit à l'utilisation des systèmes CI/CD (par exemple, travailler sur une application cloud native ou au sein d'une grande organisation). Ces données portent à croire que la culture organisationnelle et les processus de développement modernes (tels que l'intégration continue) sont les principaux moteurs de la sécurité du développement d'applications d'une organisation, et constituent le point de départ idéal pour les organisations qui cherchent à renforcer leur stratégie de sécurité.

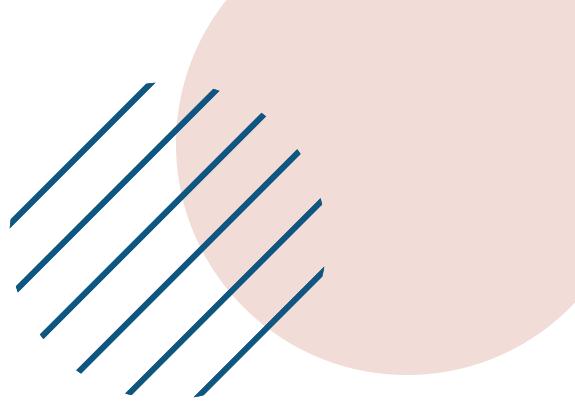
# Quels sont les bénéfices de l'adoption de bonnes pratiques de sécurité ?

À mesure que les organisations améliorent leurs pratiques de sécurité en termes de développement de logiciels, à quels avantages peuvent-elles s'attendre ? Selon les données de notre enquête, les participants estiment que **le risque de brèches de sécurité, d'interruptions de service et de dégradation des performances s'amenuise à mesure que les entreprises renforcent la mise en place de pratiques de sécurité sur la chaîne d'approvisionnement.** De même, selon d'autres études menées au cours du premier semestre 2022, l'exécution d'outils permettant par exemple d'analyser les failles pendant l'intégration continue augmente considérablement la probabilité d'identifier des problèmes au niveau des dépendances logicielles. Les utilisateurs de tels outils sont presque deux fois plus susceptibles d'avoir identifié une faille de sécurité dans leur propre code ou dans l'une de ses dépendances. En résumé : les pratiques SLSA et SSDF semblent fonctionner comme prévu. Nous ne prétendons

pas qu'elles peuvent éradiquer les menaces de sécurité, mais les résultats de notre étude suggèrent qu'elles réduisent les risques de sécurité pour les organisations.

Les pratiques de sécurité peuvent également avoir un impact positif sur les résultats basés sur les performances, mais il y a un hic : l'intégration continue joue un rôle décisif. Lorsqu'aucun système CI n'est en place, les pratiques de sécurité n'ont pas d'effets sur les performances de livraison de logiciels. Au contraire, lorsqu'un système CI est disponible, ces pratiques ont un impact positif significatif sur ces performances. Cela signifie essentiellement que l'utilisation de la CI est nécessaire pour que les pratiques de sécurité bénéficient aux performances de livraison de logiciels. Par ailleurs, les pratiques de sécurité ont généralement un effet positif sur les performances organisationnelles, et lorsque le système CI est solidement établi, cet effet est amplifié. Le graphique ci-dessous tente de représenter cet impact.





Outre la réduction des risques de sécurité perçus, les personnes interrogées ont également signalé une **réduction du burnout** parmi les membres de l'équipe et une volonté accrue de recommander leur organisation comme **un lieu de travail idéal**. Ces deux résultats témoignent de la nature ambivalente de la sécurité pour les ingénieurs logiciels : c'est un élément de plus dans leurs listes de tâches déjà bien remplies. Les outils et les processus d'intégration des pratiques de sécurisation dans les flux de développement existants leur permettent de réduire les risques de sécurité et de travailler plus sereinement, sans les contraintes des tâches imprévues ou des corrections en urgence lorsqu'une menace est découverte.

Dans l'ensemble, les résultats de l'étude suggèrent que des **pratiques de sécurité bien établies sont plus répandues dans les équipes performantes et à culture saine** (même si, comme indiqué précédemment, il existe toujours une marge d'amélioration). Les approches telles que les frameworks SLSA ou SSDF ne permettent sans doute pas à elles seules d'améliorer l'ensemble des métriques suivies concernant la culture et les performances, mais il est clair que la sécurité ne doit pas cannibaliser les autres priorités de développement.

**Les outils et les processus d'intégration des pratiques de sécurisation dans les flux de développement existants leur permettent de réduire les risques de sécurité et de travailler plus sereinement, sans les contraintes des tâches imprévues ou des corrections en urgence lorsqu'une menace est découverte.**

# 05

## Résultats inattendus

Même si le rapport que nous publions tous les ans est consacré aux réponses à l'enquête de l'année correspondante, nous faisons de notre mieux pour analyser ces résultats dans le contexte de l'ensemble du catalogue des rapports sur l'état du DevOps et des études connexes (par exemple, les études sur le burnout et sur la culture). L'un des principes fondamentaux du programme de recherche consiste à tester la fiabilité de ces effets à l'aide d'efforts de réPLICATION. Cela nous donne la possibilité d'ajuster nos hypothèses en fonction des données et d'analyser les tendances fluctuantes ou émergentes.

Cette année, certains résultats sont surprenants. Cela peut s'expliquer de nombreuses façons. Pour commencer, l'échantillon du rapport 2022 inclut davantage de personnes en début de carrière que les rapports précédents. Voici l'une des interprétations possibles : les personnes qui sont directement responsables de la mise en œuvre des pratiques et des capacités techniques s'expriment davantage que les personnes potentiellement chargées de superviser la

mise en œuvre de ces pratiques. Autre possibilité : le secteur ou le monde a évolué. Ce qui a fonctionné hier ne fonctionne pas forcément aujourd'hui. Par exemple, les forces macroéconomiques ainsi qu'une nouvelle année largement dominée par la pandémie de COVID-19 ont peut-être changé l'univers du DevOps. Enfin, les légères modifications apportées à la composition de notre modèle ont probablement influencé les relations entre les variables<sup>1</sup>.



Derek DeBellis



<sup>1</sup> Les ouvrages "Book of Why" de Judea Pearl et "Statistical Rethinking" de Robert McElreath's illustrent via des exemples étonnantes la façon dont les résultats d'un modèle statistique peuvent varier selon ce qu'on y inclut ou non.

Des résultats inattendus ou sans hypothèse associée mettent les chercheurs dans une situation délicate lors de la rédaction du rapport. Compte tenu du risque de faux résultats ou, tout du moins, de résultats en attente de confirmation (voire de réfutation) des preuves empiriques recueillies via plusieurs études, l'attitude la plus responsable consiste à effectuer des études complémentaires pour tenter de reproduire les résultats et d'en comprendre la cause<sup>2</sup>. En se concentrant sur les résultats inattendus, les chercheurs courrent aussi le risque de dévaloriser les résultats établis de manière fiable au cours de plusieurs années de recherche. Pour chaque rapport sur l'état du DevOps, notre exploration statistique porte sur plus d'une centaine d'hypothèses. Ce faisant, nous prenons le risque que le hasard provoque de faux résultats. Nous essayons de contrecarrer cela à l'aide d'efforts de réplication annuels.

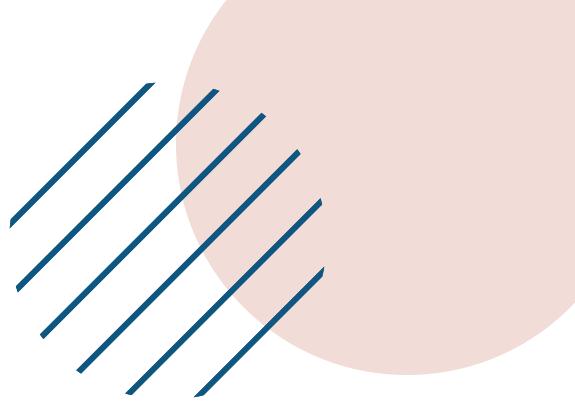
Toutefois, le fait de ne pas publier les résultats risque de créer un effet tiroir<sup>3</sup>, qui consiste à ne diffuser que les résultats attendus ou acceptables, tandis que les résultats inattendus ou difficiles à accepter restent cachés. Tout est question d'équilibre : nous voulons éviter le sensationnalisme autour des résultats précoces, tout en étant convaincus qu'il est important de les partager. Voici les résultats qui nous ont le plus surpris, ainsi que notre interprétation :

**01** Historiquement, nous observons que les pratiques de développement à branche unique ont un impact positif sur les performances de livraison de logiciels. Cette observation se vérifie dans chaque étude depuis 2014. En 2022, les résultats sur les capacités de développement à branche unique sont hors normes. D'une part, les fonctionnalités à branche unique ont un impact négatif sur les performances de livraison de logiciels. Les recherches précédentes montraient l'inverse. Étant donné le degré d'absurdité de ce résultat, nous avons hâte de voir s'il se reproduit dans les recherches à venir et de recueillir les pistes d'explication de la communauté.

**02** Nous constatons que les performances de livraison de logiciels ne bénéficient aux performances organisationnelles que lorsque les performances opérationnelles sont également élevées. De plus, de nombreuses personnes interrogées ne déclarent pas de performances opérationnelles élevées. Cela contredit les éditions précédentes de notre étude, dans lesquelles le lien entre les performances de livraison de logiciels et les performances organisationnelles était beaucoup plus clair.

<sup>2</sup> Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and social psychology review*, 2(3), 196-217.

<sup>3</sup> Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological bulletin*, 86(3), 638.



**03** Les pratiques de documentation ont un impact négatif sur les performances de livraison de logiciels. Ce résultat contredit les rapports précédents. Une hypothèse est que la documentation devient une pratique de plus en plus automatisée, en particulier parmi les équipes performantes. Sans données supplémentaires, nous ne sommes pas en mesure de valider ou de réfuter cette hypothèse.

**04** Certaines capacités techniques (développement à branche unique, architecture faiblement couplée, CI/CD, etc.) semblent constituer des indicateurs de burnout. Comme mentionné ci-dessus, de nombreux participants de l'échantillon 2022 étaient nettement moins avancés dans leur carrière que ceux des échantillons des années précédentes. Par conséquent, il est possible que nous ayons échangé avec les personnes responsables de la mise en œuvre et non avec celles responsables de la création ou de la supervision des initiatives. Le processus de mise en œuvre peut être nettement plus difficile que la supervision. Nous souhaitons mener d'autres recherches pour mieux comprendre ce résultat.



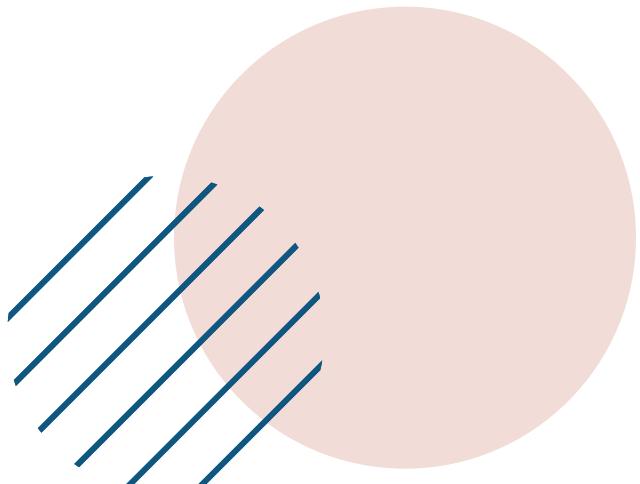
**05** Les pratiques d'ingénierie en fiabilité ont un impact négatif sur les performances de livraison de logiciels, mais il n'y a pas nécessairement de lien de causalité entre les deux. Selon une nouvelle analyse par regroupement réalisée cette année (voir "Où en êtes-vous ?"), un sous-ensemble de groupes semble se focaliser sur la fiabilité sans tenir compte des performances de livraison de logiciels. Nous pensons que ces deux objectifs sont dissociés (en d'autres termes, vous pouvez en atteindre un sans atteindre l'autre). Cependant, pour que les performances de livraison de logiciels puissent impacter les performances organisationnelles, la fiabilité est indispensable.

**06** Nous avons ajouté des pratiques liées au framework SLSA pour déterminer si les équipes adoptent ces approches afin d'assurer la sécurité sur la chaîne d'approvisionnement logicielle. Nous nous attendions à ce qu'un lien existe entre la mise en œuvre des pratiques de sécurité et les performances (utilisation de capacités techniques, meilleures performances de livraison de logiciels et organisationnelles, etc.), mais nous avons été étonnés de constater que les capacités techniques s'appuient sur ces pratiques de sécurité pour impacter les performances de livraison de logiciels et organisationnelles.

L'inclusion de pratiques liées au SLSA semble expliquer en grande partie les effets de l'intégration continue, du contrôle des versions et de la livraison continue sur les performances de livraison de logiciels et organisationnelles. En d'autres termes, une chaîne causale détectée dans les données indique que de nombreuses capacités techniques ont un impact positif sur les pratiques liées au framework SLSA, qui impactent à leur tour les performances de livraison de logiciels et les performances organisationnelles. Pour déterminer ce résultat, nous avons eu recours à des analyses de médiation<sup>45</sup>. Celles-ci nous invitent à déterminer si la mesure des pratiques liées au framework SLSA englobe le suivi d'autres caractéristiques de l'équipe (ses performances générales, par exemple) et de quelle manière les pratiques de sécurité conduisent à de meilleures performances de livraison de logiciels et organisationnelles.

Nous avons hâte d'étudier à nouveau ces effets l'année prochaine afin de déterminer si ces schémas inédits se reproduisent ou s'il peuvent être considérés comme des anomalies (devant toutefois être expliquées). Comme toujours, n'hésitez pas à nous faire part de vos commentaires.

**Rejoignez la communauté DORA (<http://dora.community>) pour poursuivre la discussion sur ces résultats inattendus et sur les autres informations du rapport 2022.**



<sup>4</sup> Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data", Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

<sup>5</sup> Carrión, Gabriel Cepeda, Christian Nitzl & José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples". Partial least squares path modeling. Springer, Cham, 2017. 173-195.

# 06

## Données sur les entreprises et démographiques



Derek DeBellis

Nous tenons à vous remercier pour vos précieuses contributions à notre enquête et au secteur en général.

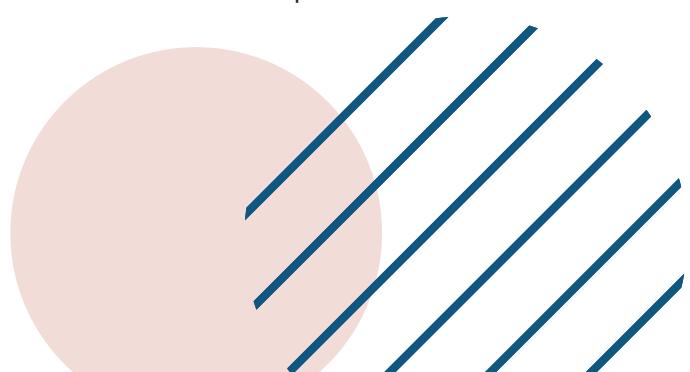
### Qui a répondu à l'enquête ?

Basé sur huit ans de recherches et plus de 33 000 réponses recueillies auprès de professionnels du secteur, le rapport sur l'état du DevOps décrit les pratiques de développement logiciel et de DevOps qui favorisent la réussite des équipes et des entreprises. Cette année, plus de 1 350 professionnels de divers secteurs à travers le monde ont partagé leurs expériences pour nous aider à mieux comprendre les facteurs à l'origine de meilleures performances. Nous tenons à les remercier pour leurs précieuses contributions à notre enquête et au secteur en général. En résumé, la représentation tirée des diverses données

démographiques et sur les entreprises est restée remarquablement cohérente.

Comme les années précédentes, nous avons recueilli les données démographiques de chaque personne qui a répondu à l'étude. Les catégories sont les suivantes : le genre, le handicap et les groupes sous-représentés.

Cette année, nous avons constaté une représentation cohérente avec les rapports précédents dans les catégories se rapportant à l'entreprise, comme la taille, le secteur d'activité et la région. À nouveau, plus de 60 % des personnes interrogées occupent des postes d'ingénieurs ou de responsables, et un tiers travaillent dans le secteur de la technologie. En outre, les secteurs des services financiers, du commerce et des entreprises industrielles et manufacturières sont représentés.



# Données démographiques

## Genre

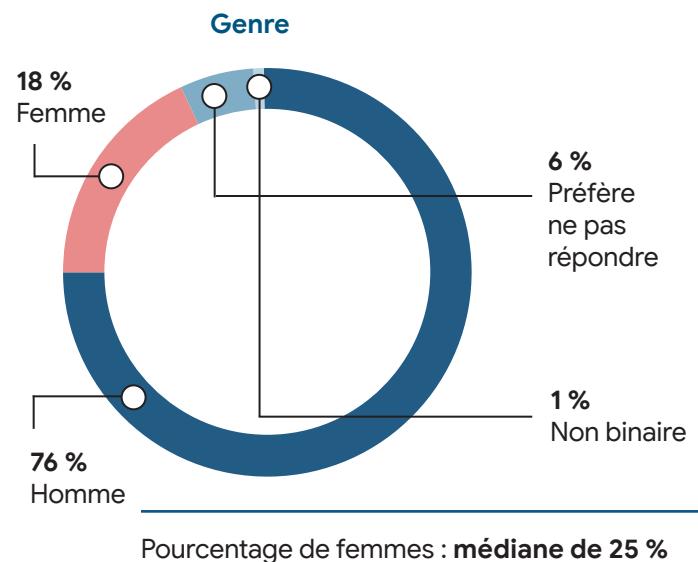
L'échantillon de personnes interrogées cette année comptait une proportion plus élevée de femmes qu'en 2021 (18 % contre 12 %), tandis que la proportion d'hommes était plus faible (76 % contre 83 %). Les personnes interrogées ont déclaré que les femmes représentaient environ 25 % de leurs équipes, une proportion identique à celle de 2021 (25 %).

## Handicap

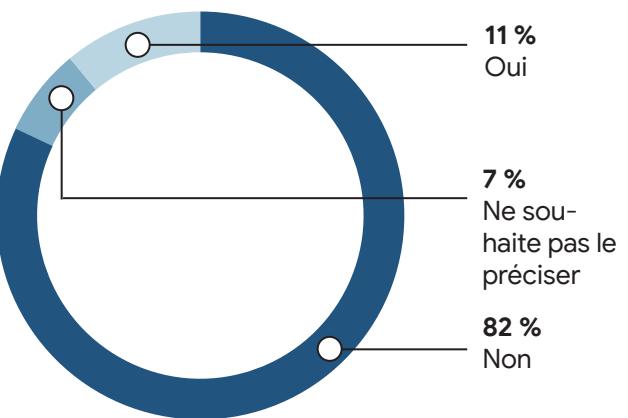
Nous avons identifié le handicap selon six aspects correspondant aux directives du [Washington Group Short Set](#). Nous posons des questions sur le handicap pour la quatrième année. Le pourcentage de personnes ayant un handicap était cohérent avec notre rapport de 2021, soit 11 %.

## Sous-représentation

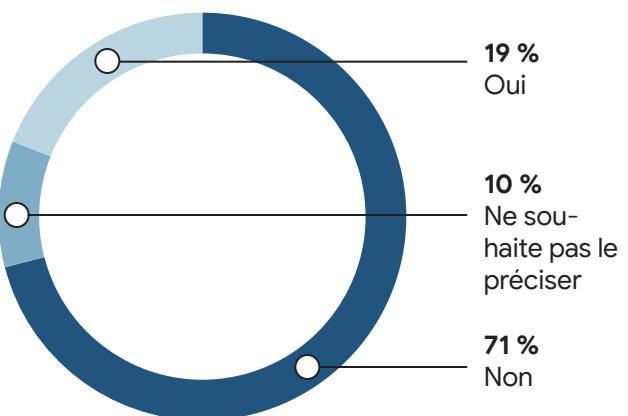
S'identifier comme membre d'un groupe sous-représenté peut faire référence au groupe ethnique, au genre ou à une autre caractéristique. Nous posons des questions sur la sous-représentation pour la cinquième année. Le pourcentage d'individus qui s'identifient comme sous-représentés a légèrement augmenté, passant de 17 % en 2021 à 19 % en 2022.



## Handicap



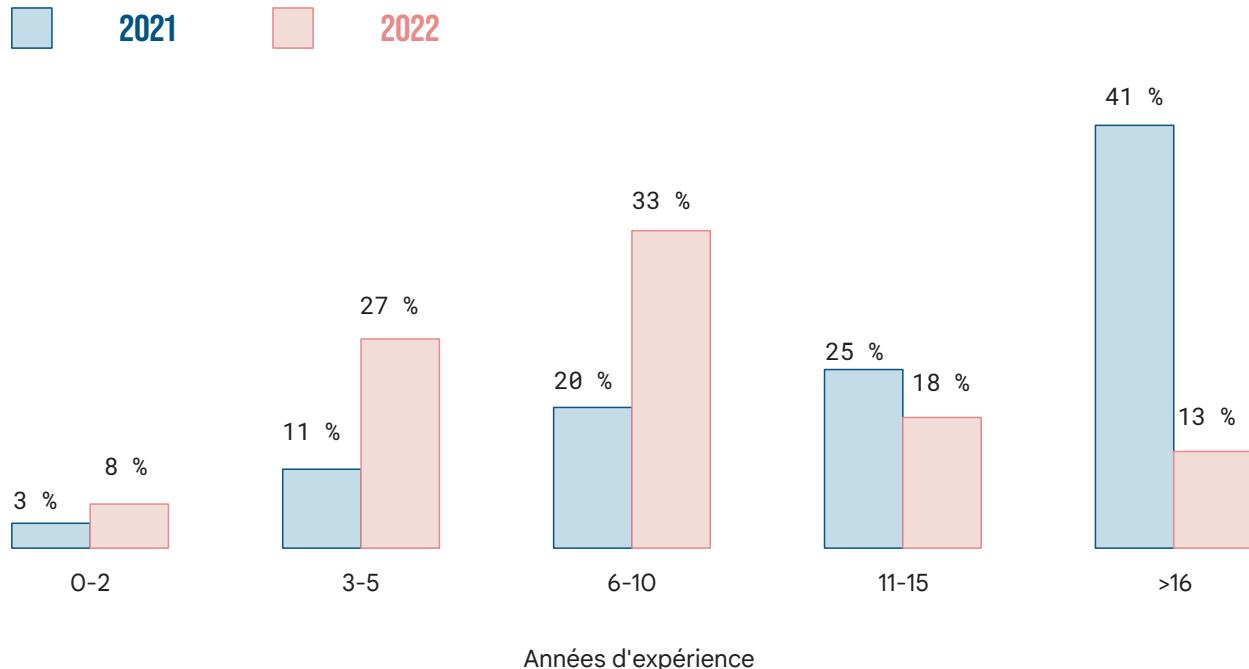
## Sous-représentation

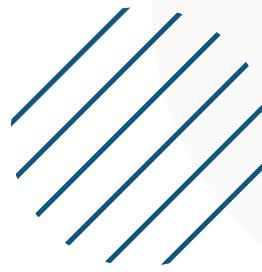


## Années d'expérience

Cette année, 35 % des participants à l'enquête avaient cinq ans ou moins d'expérience professionnelle, une proportion nettement supérieure à 2021 (14 %). Sans surprise, beaucoup moins de participants avaient 16 ans ou plus d'expérience par rapport à 2021 (13 % contre 41 %).

Cette évolution peut expliquer certaines tendances ayant émergé dans les données, et nous pensons qu'il est important de garder cela à l'esprit lors de l'interprétation des résultats, surtout si on les compare avec ceux de l'année dernière.

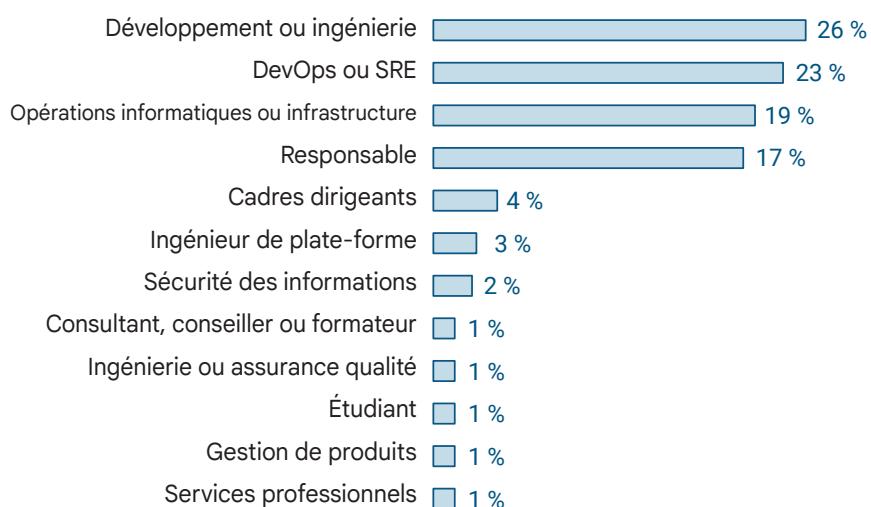




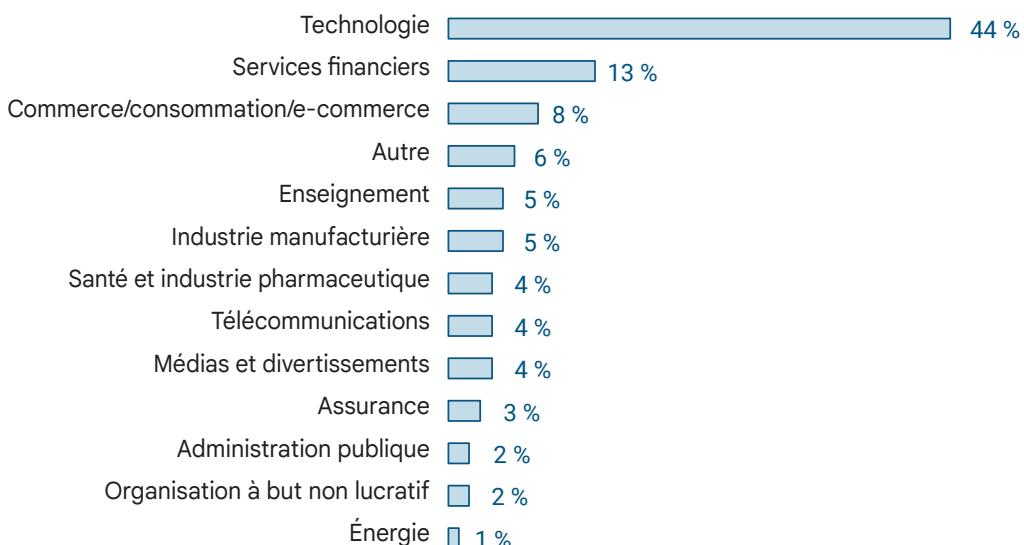
## Données sur les entreprises

### Rôle

85 % des participants travaillent au sein d'une équipe de développement ou d'ingénierie (26 %), d'une équipe DevOps ou SRE (23 %), d'un service informatique ou en charge de l'infrastructure (19 %) ou sont responsables (17 %). La proportion de participants qui travaillent dans des services informatiques ou en charge de l'infrastructure a plus que doublé par rapport à l'année dernière, passant de 9 % à 19 %. Les deux catégories les plus en baisse par rapport à l'année dernière sont les cadres dirigeants (de 9 % à 4 %), et les services professionnels (de 4 % à 1 %).



### 2022



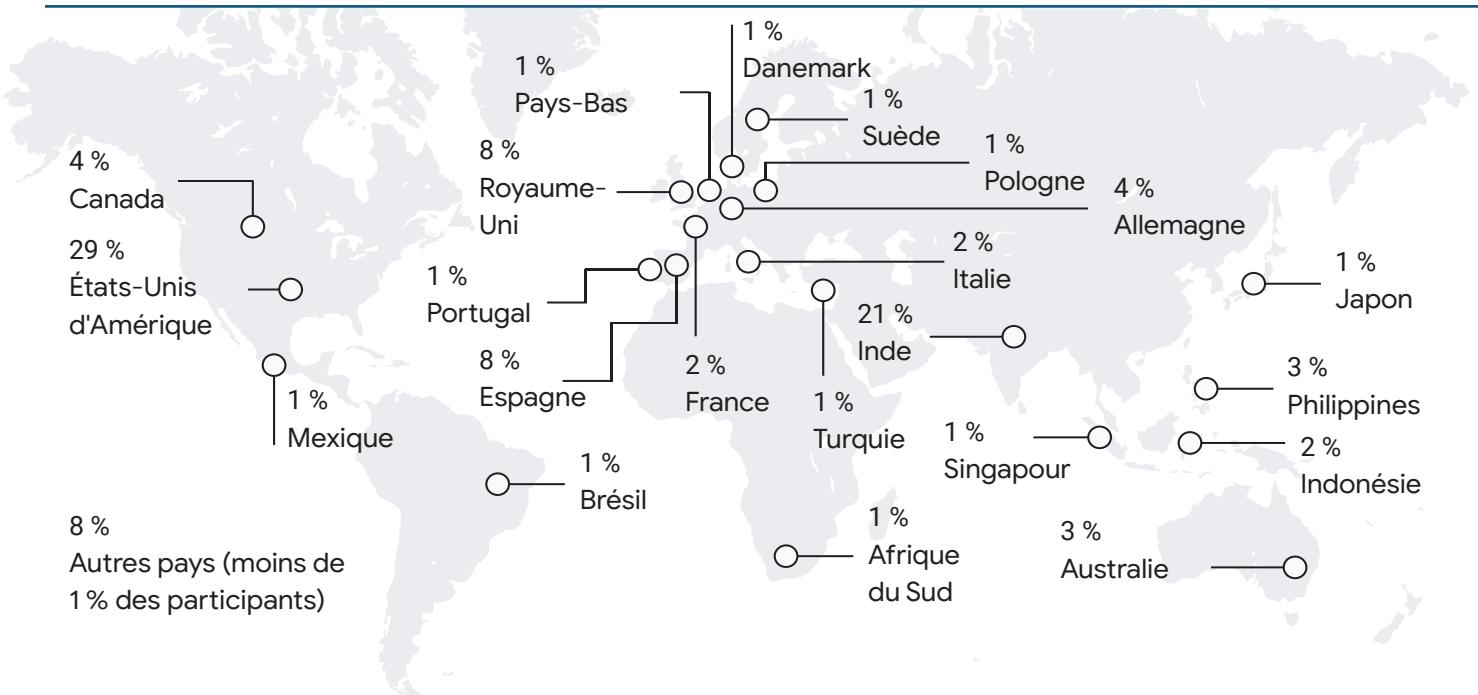
### Secteur

Comme dans les rapports sur l'état du DevOps précédents, nous constatons que la plupart des personnes interrogées travaillent dans le secteur de la technologie, suivi par les services financiers, d'autres secteurs, puis le commerce.

## Région

Cette année, nous avons demandé aux participants de sélectionner leur pays d'origine plutôt que leur région.

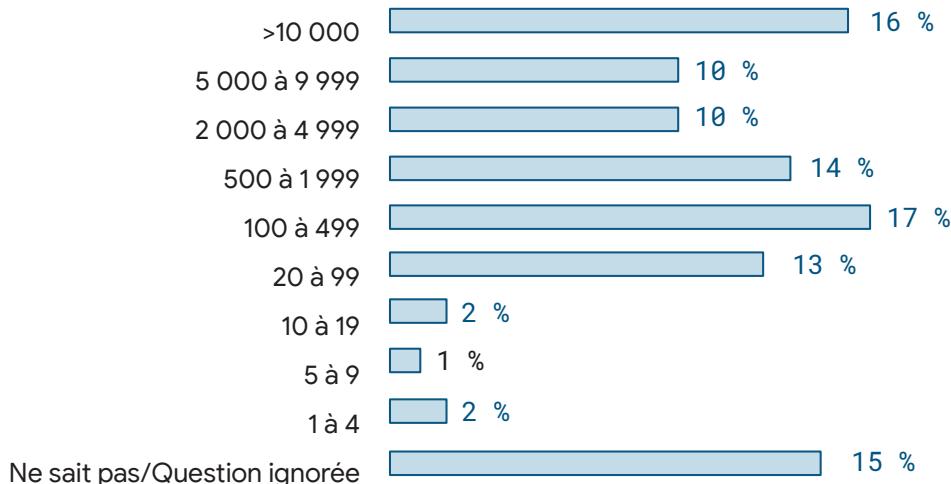
Les régions, souvent représentées par le nom d'un continent, semblaient un peu trop vagues pour permettre l'analyse du profil des participants. Nous avons reçu des réponses provenant de plus de 70 pays, avec une concentration de 89 % des participants dans 22 pays.

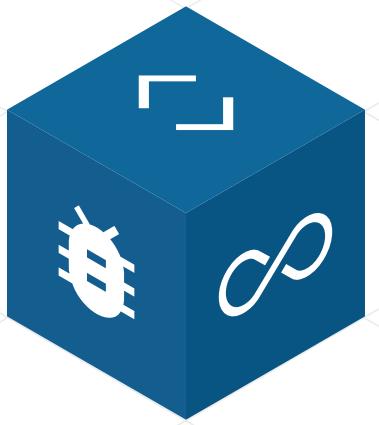


## Nombre d'employés

Conformément aux précédentes enquêtes sur l'état du DevOps, les participants proviennent d'organisations de tailles diverses. Environ 22 % d'entre eux travaillent dans des organisations de plus de 10 000 employés, 7 % dans des entreprises comptant entre 5 000 et 9 999 employés et 15 % dans des organisations comptant entre 2 000 et

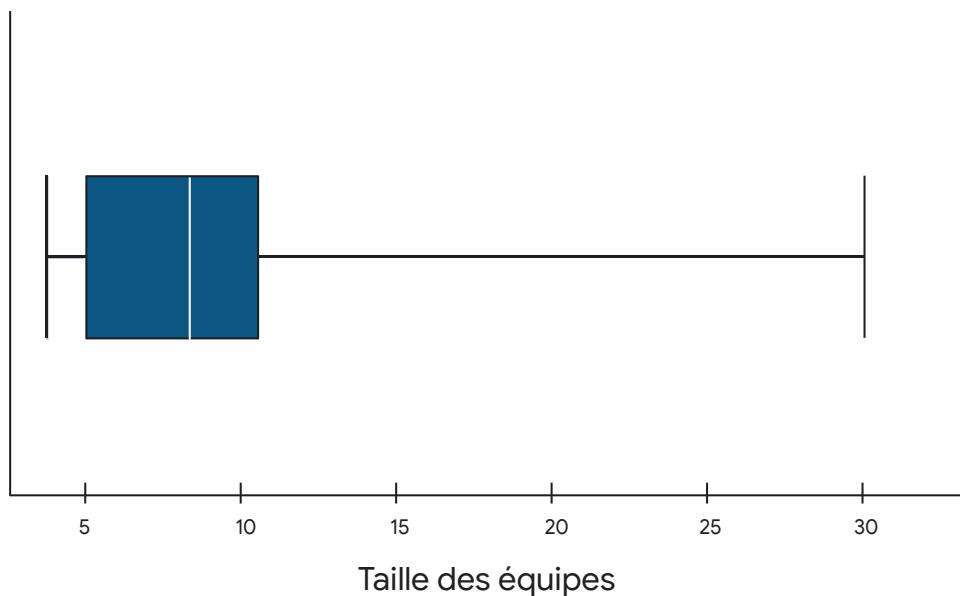
4 999 employés. Nous avons également constaté une bonne représentation de personnes issues d'organisations comptant 500 à 1 999 employés (13 %), 100 à 499 employés (15 %) et 20 à 99 employés (15 %). Cette année, les participants pouvaient également sélectionner l'option "Je ne sais pas" concernant la taille de leur organisation, et 15 % d'entre eux ont déclaré ne pas la connaître ou ont ignoré la question.





## Taille des équipes

Cette année, nous avons demandé aux participants d'indiquer le nombre approximatif de personnes dans leur équipe. 25 % des participants ont indiqué travailler dans des équipes de cinq personnes maximum, 50 % dans des équipes de huit personnes maximum et 75 % dans des équipes de 12 personnes maximum.

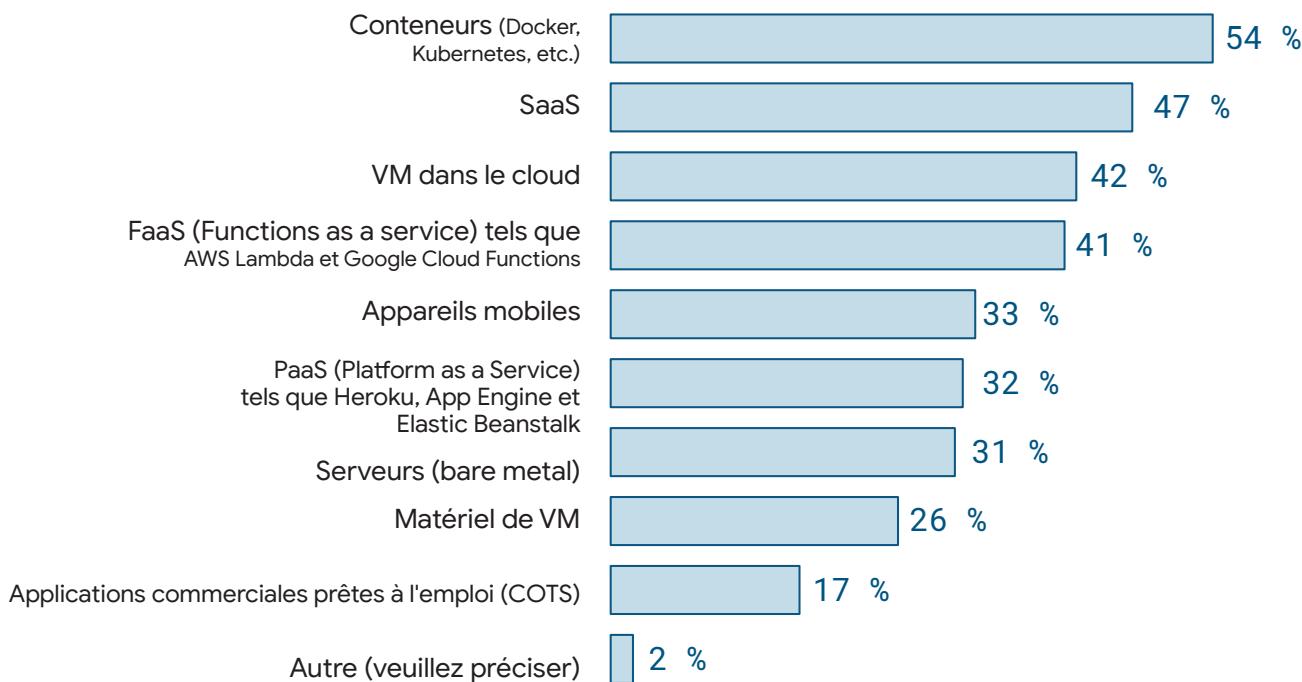


## Cible de déploiement

Pour la première fois en 2021, nous avions décidé d'étudier l'emplacement de déploiement de l'application ou du service principal sur lequel les participants travaillent. À notre grande surprise, les conteneurs étaient la première cible de déploiement. Cela s'est confirmé cette

année, mais dans une moindre mesure (54 % contre 64 % en 2021).

Nous avons également ajouté davantage d'options afin que les participants puissent indiquer l'emplacement de leurs déploiements de manière plus précise.



# 07

## Conclusions



Derek DeBellis

Chaque fois que nous produisons ce rapport, nous nous efforçons de fournir un compte rendu rigoureux de la façon dont les pratiques et les capacités conduisent à des résultats essentiels pour les entreprises, comme les performances organisationnelles. Nous évaluons la réplicabilité de nombreux effets constatés dans les rapports précédents et élargissons la portée de notre travail pour tenir compte des priorités émergentes dans l'espace DevOps. Cette année, nous avons façonné notre enquête et nos analyses de manière à approfondir l'étude des pratiques de sécurité, et nous avons modifié notre approche de modélisation statistique afin d'explorer la conditionnalité ou les dépendances de certains effets. Nous avons également exploré de nouvelles façons de décrire le paysage des performances de livraison de logiciels et des performances opérationnelles.

À bien des égards, le récit qui se matérialise cette année fait écho aux résultats des années précédentes : les capacités techniques s'appuient les unes sur les autres pour augmenter les performances, l'utilisation du cloud présente de nombreux avantages, la culture et la flexibilité de l'environnement de travail conduisent à des performances organisationnelles plus élevées, et le burnout des employés empêche les organisations d'atteindre leurs objectifs. Grâce à l'analyse des interactions que nous avons explicitement ajoutée au modèle, nous avons pu comprendre les conditions dans lesquelles certains effets se produisent.

Par exemple, les performances de livraison de logiciels ne semblent avoir un impact positif sur les performances organisationnelles que lorsque les performances opérationnelles (c'est-à-dire la fiabilité) sont élevées, ce qui nous amène à conclure que vous avez besoin des deux pour réussir en tant qu'organisation. En outre, certains résultats sont inattendus (voir la section dédiée).

Nous remercions toutes les personnes qui ont contribué à l'enquête de cette année, et espérons que nos recherches vous aideront ainsi que votre entreprise à créer des équipes et des logiciels plus performants (tout en préservant l'équilibre travail-vie personnelle).

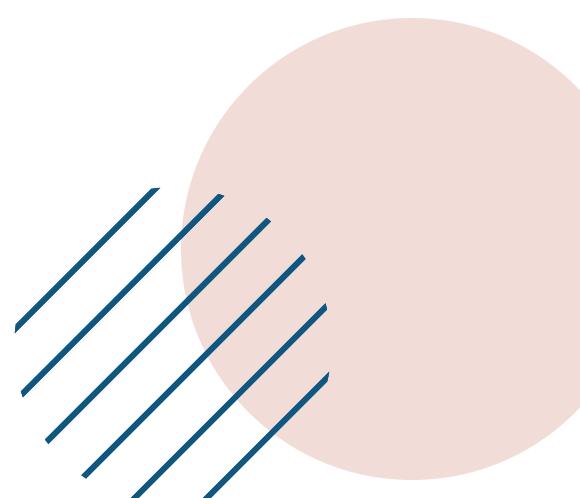


# 08

## Remerciements

Ce rapport 2022 n'aurait pas vu le jour sans l'aide d'une grande famille de contributeurs passionnés. Nos collègues nous ont aidés à réaliser ce travail considérable notamment par l'élaboration des questions, l'analyse, la rédaction, la correction et la conception du rapport. Les auteurs aimeraient remercier toutes les personnes qui leur ont fait part de leurs commentaires et conseils sur le rapport tout au long de l'année. Tous les remerciements sont formulés par ordre alphabétique.

Scott Aucoin	Eric Maxwell
Alex Barrett	John Speed Meyers
James Brookbank	Steve McGhee
Kim Castillo	Jacinda Mein
Lolly Chessie	Alison Milligan
Jenna Dailey	Pablo Pérez Villanueva
Derek DeBellis	Claire Peters
Rob Edwards	Connor Poske
Dave Farley	Dave Stanke
Christopher Grant	Dustin Smith
Mahshad Haeri	Seth Vargo
Nathen Harvey	Daniella Villalba
Damith Karunaratne	Brenna Washington
Todd Kulesza	Kaiyuan "Frank" Xu
Amanda Lewis	Nicola Yap
Ian Lewis	



# 09

## Auteurs



### Claire Peters

Claire Peters est chercheuse en expérience utilisateur chez Google. Ses recherches visent à appliquer divers aspects et idées de DORA sur le terrain. Elle étudie le programme de modernisation des applications cloud (CAMP) de Google, les engagements client axés sur DORA et les outils liés aux quatre indicateurs clés, dans le but d'aider les équipes et les individus à appliquer plus efficacement les principes DORA dans leur travail quotidien. Claire est également membre de l'équipe de recherche principale de DORA et contribue à l'enquête annuelle DORA ainsi qu'au rapport sur l'état du DevOps. Elle est titulaire d'un master en analyse culturelle appliquée de l'Université de Copenhague.



### Dave Farley

Dave Farley est directeur général et fondateur de Continuous Delivery Ltd, et créateur de la chaîne YouTube Continuous Delivery. Il est respectivement co-auteur et auteur des livres à succès "Continuous Delivery" et "Modern Software Engineering: Doing What Works to Build Better Software Faster". Il a co-écrit "Reactive Manifesto" et est lauréat du Duke Award pour le projet Open Source "LMAX Disruptor". Dave est un pionnier de la livraison continue, un leader d'opinion et un praticien expert en CD, DevOps, TDD et en conception de logiciels. Il possède une longue expérience dans la création d'équipes hautement performantes, l'accompagnement d'organisations sur la voie de la réussite et la création de logiciels exceptionnels. Retrouvez Dave sur [Twitter](#), [YouTube](#), son [blog](#) et son [site Web](#).



## Daniella Villalba

Daniella Villalba est chercheuse en expérience utilisateur et se consacre au projet DORA. Elle s'attache à comprendre les facteurs qui rendent les développeurs heureux et productifs. Avant de travailler pour Google, Daniella a étudié les avantages de la formation à la méditation, les facteurs psychosociaux qui affectent les expériences des étudiants, la mémoire des témoins oculaires et les faux aveux. Elle est diplômée d'un doctorat en psychologie expérimentale de l'université internationale de Floride.



## Dave Stanke

Dave Stanke est responsable des relations avec les développeurs chez Google, où il conseille les clients sur les pratiques facilitant l'adoption du DevOps et de la SRE. Tout au long de sa carrière, il a porté de nombreuses casquettes, y compris directeur de la technologie d'une start-up, responsable produit, membre du service client, développeur logiciel, administrateur système et graphiste. Il est titulaire d'un master en gestion de la technologie de l'université de Columbia.



## Derek DeBellis

Derek DeBellis est chercheur en expérience utilisateur quantitative chez Google. Il se consacre aux recherches basées sur des enquêtes, à l'analyse des journaux et à la recherche de solutions pour mesurer les principaux concepts du développement de produits. Derek a récemment publié des articles sur les interactions humain/IA, sur l'impact de l'apparition de la Covid-19 sur l'arrêt du tabac, sur la conception dédiée aux problèmes de TLN et sur le rôle de l'expérience utilisateur dans les débats sur la confidentialité.



## Eric Maxwell

Eric Maxwell est responsable de l'unité Google chargée de la transformation numérique basée sur le DevOps, où il aide les plus grandes entreprises du monde à s'améliorer de façon progressive et continue. Ancien ingénieur sur le terrain, Eric a consacré la première moitié de sa carrière à automatiser tout ce qu'il pouvait afin d'aider ses pairs. Il a co-créé le programme de modernisation des applications cloud (CAMP) de Google, est membre de l'équipe principale DORA et auteur du guide du DevOps pour les entreprises. Avant d'intégrer Google, Eric a travaillé sur la conception de solutions géniales avec ses collègues pleins d'esprit chez Chef Software.



## John Speed Meyers

John Speed Meyers est data scientist en sécurité chez Chainguard, une startup spécialisée dans la sécurité sur la chaîne d'approvisionnement logicielle. Les projets de recherche de John portent sur des sujets variés tels que la sécurité sur la chaîne d'approvisionnement logicielle, la sécurité des logiciels Open Source et la réponse internationale à la croissance de la puissance militaire chinoise. Auparavant, John a travaillé chez In-Q-Tel, la RAND Corporation et le Center for Strategic and Budgetary Assessments. John est titulaire d'un doctorat en analyse des stratégies de la Pardee RAND Graduate School, d'un master en affaires publiques de la Princeton's School of Public and International Affairs et d'une licence en relations internationales de l'université Tufts.



## Kaiyuan "Frank" Xu

Kaiyuan "Frank" Xu est chercheur en expérience utilisateur quantitative chez Google. Il analyse les données des journaux et des enquêtes afin de comprendre les habitudes et les commentaires des utilisateurs des produits Google Cloud, améliorant ainsi la qualité des produits pour les développeurs. Avant d'intégrer Google, Kaiyuan a mené pendant plusieurs années des recherches qualitatives et quantitatives sur l'expérience utilisateur pour les produits Azure et Power Platform chez Microsoft. Il est titulaire d'un master en conception et ingénierie axées sur l'humain à l'université de Washington.



## Nathen Harvey

Nathen Harvey est responsable des relations avec les développeurs chez Google. Il a construit sa carrière en aidant les équipes à réaliser leur potentiel tout en alignant la technologie sur les résultats commerciaux. Nathen a eu la chance de travailler avec certaines des meilleures équipes et communautés Open Source, les accompagnant dans l'application des principes et pratiques du DevOps et de SRE. Nathen a co-édité et contribué à l'ouvrage *97 Things Every Cloud Engineer Should Know*, O'Reilly 2020.



## Todd Kulesza

Todd Kulesza est chercheur en expérience utilisateur chez Google, où il étudie comment les ingénieurs logiciels travaillent aujourd'hui et comment ils pourraient mieux travailler demain. Il est titulaire d'un doctorat en informatique de l'université d'État de l'Oregon.

# 10

## Méthodologie

### Conception de la recherche

Cette étude utilise une conception transversale, basée sur la théorie. Également appelée **analyse prédictive inférentielle**, il s'agit de l'un des types de conception les plus couramment utilisés actuellement dans la recherche commerciale et technologique. La conception inférentielle prédictive est utilisée lorsque la conception purement expérimentale n'est pas pratique ni possible.

### Population cible et échantillonnage

La population cible pour cette enquête se compose d'utilisateurs et de responsables qui travaillent dans les secteurs de la technologie et des transformations (ou en lien étroit avec ceux-ci), en particulier ceux qui connaissent le DevOps. Nous avons fait la promotion de l'enquête via des listes de diffusion, des campagnes sur Internet, un panel en ligne, les réseaux sociaux, et en demandant aux cibles de partager l'enquête avec leurs réseaux (échantillonnage boule de neige).

### Création de constructions latentes

Nous avons formulé nos hypothèses et idées à l'aide de constructions précédemment vérifiées lorsque cela était possible. Nous avons développé de nouvelles constructions basées sur la théorie, des définitions et des avis d'experts. Nous avons ensuite réalisé des étapes supplémentaires pour clarifier l'intention afin de nous assurer que les données recueillies grâce à l'enquête soient véritablement susceptibles d'être fiables et valables<sup>1</sup>.

### Calcul des différences entre les équipes les moins performantes et les plus performantes

Dans la section "Où en êtes-vous ?", nous comparons les équipes les moins performantes et les plus performantes selon les quatre indicateurs de performances de livraison. Cette méthode est simple. Prenons l'exemple de la fréquence de déploiement. Les équipes du groupe "élevé" effectuent des déploiements à la demande (c'est-à-dire plusieurs fois par jour). Si elles effectuent en moyenne quatre déploiements par jour, cela revient à 1 460 déploiements par an ( $4 \times 365$ ). Les équipes du groupe "faible", en revanche, effectuent des déploiements entre une fois par mois et une fois tous les six mois, pour une fréquence de déploiement moyenne d'une fois tous les 3,5 mois, soit environ 3,4 déploiements par an ( $12/3,5$ ). À partir du ratio valeur haute (1 460 déploiements)/valeur basse (3,4 déploiements), nous obtenons un coefficient de 417. Cette approche est généralisée pour les autres métriques de performance de développement.

<sup>1</sup> Churchill Jr, G. A. "A paradigm for developing better measures of marketing constructs," Journal of Marketing Research 16:1, (1979), 64–73.

# Méthodes d'analyse statistique

## Analyse par groupe

Pour les deux solutions de regroupement décrites dans la section "Où en êtes-vous?", nous nous sommes appuyé sur le regroupement hiérarchique et sur une version de la méthode agglomérative de Ward<sup>2</sup> pour évaluer dans quelle mesure différentes solutions de regroupement correspondaient aux données.

Pour les premiers résultats de regroupement présentés, nous avons recherché des groupes de réponses en fonction de la fréquence de déploiement, du délai d'implémentation, du délai de restauration d'un service et du taux d'échecs sur l'implémentation de modifications. Cette année, après avoir évalué 14 solutions de regroupement hiérarchique différentes à l'aide de 30 indices différents permettant de déterminer le nombre de groupes<sup>3</sup>, nous en avons identifié trois.

La deuxième analyse par regroupement présentée était identique à la première d'un point de vue méthodologique, mais déployée selon des dimensions différentes de données. Notre but était de rechercher des modèles de réponse communs (c'est-à-dire des

groupes) en termes de productivité (combinaison de la fréquence de déploiement et du délai d'implémentation), de performances opérationnelles (fiabilité) et de stabilité (combinaison du délai de restauration des services et du taux d'échecs sur l'implémentation de modifications). Nous avons également exploré différents algorithmes de regroupement pour déterminer dans quelle mesure nos résultats étaient sensibles à notre approche. Bien qu'aucun moyen établi ne permette de quantifier cette sensibilité (à notre connaissance), les groupes identifiés présentent des caractéristiques similaires.

## Modèle de mesure

Avant d'effectuer l'analyse, nous avons identifié des constructions en utilisant l'analyse factorielle exploratoire et l'analyse en composantes principales avec la méthode de rotation varimax<sup>4</sup>. Nous avons confirmé les tests statistiques de validité et fiabilité convergentes et divergentes avec la variance moyenne extraite (AVE, Average Variance Extracted), la corrélation, le coefficient alpha de Cronbach<sup>5</sup>, le rhoA<sup>6</sup>, le rapport hétérotrait-monotrait<sup>5,7</sup> et la fiabilité composite.

<sup>2</sup> Murtagh, Fionn & Pierre Legendre. "Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?" Journal of classification 31.3 (2014): 274-295.

<sup>3</sup> Charrad M., Ghazzali N., Boiteau V., Niknafs A. (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set.", Journal of Statistical Software, 61(6), 1-36., URL <http://www.jstatsoft.org/v61/i06/>

<sup>4</sup> Straub, D., Boudreau, M. C., & Gefen, D. (2004). Validation guidelines for IS positivist research. Communications of the Association for Information systems, 13(1), 24.

<sup>5</sup> Nunnally, J.C. Psychometric Theory. New York: McGraw-Hill, 1978

<sup>6</sup> Hair Jr, Joseph F., et al. "Partial least squares structural equation modeling (PLS-SEM) using R: A workbook." (2021): 197.

<sup>7</sup> Brown, Timothy A. & Michael T. Moore. "Confirmatory factor analysis." Handbook of structural equation modeling 361 (2012): 379.

## Modélisation par équation structurelle

Nous avons testé les modèles d'équation structurelle (SEM, Structural Equation Models) en utilisant l'analyse de régression par moindres carrés partiels<sup>8</sup>, qui est un SEM basé sur la corrélation.

## Analyse du deuxième modèle de regroupement

Pour comprendre les indicateurs d'appartenance à un groupe, nous avons utilisé la régression logistique multinomiale<sup>9</sup>. En effet, notre objectif était de prédire l'appartenance à un groupe, qui, dans ce cas, est constituée de données catégorielles non ordonnées à plus de deux niveaux. Pour interpréter les résultats obtenus selon l'appartenance à un groupe, nous nous sommes appuyés sur une régression linéaire pour chacun de d'entre eux (burnout, tâches imprévues et performances organisationnelles).

<sup>8</sup> Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2021). "A primer on partial least squares structural equation modeling (PLS-SEM)." Sage publications.

<sup>9</sup> Ripley, Brian, William Venables (maintenance : Brian Ripley). "Package 'nnet'." R package version 7.3-12 (2016): 700.

# 11

## Documentation complémentaire

Rejoignez la communauté DORA pour discuter, apprendre et collaborer sur l'amélioration des performances de livraison de logiciels et opérationnelles.

<http://dora.community>

Apprenez-en plus sur les quatre indicateurs clés.

<https://goo.gle/four-keys>

Obtenez plus d'informations sur les capacités DevOps.

<https://goo.gle/devops-capabilities>

Découvrez comment mettre en œuvre les pratiques DORA au sein de votre organisation dans notre guide pour les entreprises.

<https://goo.gle/enterprise-guidebook>

Obtenez des ressources sur l'ingénierie en fiabilité des sites (SRE).

<https://sre.google>

<https://goo.gle/enterprise-roadmap-sre>

Participez à l'évaluation rapide DevOps.

<https://goo.gle/devops-quickcheck>

Découvrez le programme de recherche DevOps.

<https://goo.gle/devops-research>

Lisez l'e-book primé aux DevOps Awards de Google Cloud pour découvrir les témoignages d'autres entreprises ayant mis en œuvre les pratiques de DORA.

<https://goo.gle/devops-awards>

Découvrez le programme de modernisation des applications de Google Cloud (CAMP).

<https://goo.gle/3daLa9s>

Exploiter les données et les métriques de DORA pour transformer les processus technologiques (en anglais)

<https://goo.gle/3Doh8Km>

Lisez le livre blanc "The ROI of DevOps Transformation: How to quantify the impact of your modernization initiatives" par Forsgren, N., Humble, J., et Kim, G. (2018).

<https://goo.gle/3qECIIh>

Consultez le livre : Accelerate: *The science behind devops: Building and scaling high performing technology organizations.* IT Revolution.

<https://itrevolution.com/book/accelerate>

Apprenez-en plus sur le framework des niveaux de la chaîne d'approvisionnement pour les artefacts logiciels (SLSA).

<https://slsa.dev>

Apprenez-en plus sur le framework de développement de logiciels sécurisés (framework NIST SSDF).

<https://goo.gle/3qBXLWk>

Apprenez-en plus sur la culture DevOps : Culture organisationnelle Westrum.

<https://goo.gle/3xq7KBV>

Apprenez-en plus sur l'Open Source Security Foundation. <https://openssf.org/>

Apprenez-en plus sur in-toto.

<https://in-toto.io/>

Apprenez-en plus sur la nomenclature de NTIA.gov.

<https://www.ntia.gov/SBOM>

Cybersecurity: Federal Response to SolarWinds and Microsoft Exchange Incidents

<https://www.gao.gov/products/gao-22-104746>

Apprenez-en plus sur l'analyse de la sécurité au niveau de l'application dans un cadre CI/CD.

<https://go.dev/blog/survey2022-q2-results#security>

Enfin, et surtout, pensez à consulter les rapports précédents sur l'état du DevOps. Ils sont tous disponibles à l'adresse <https://goo.gle/dora-sodrs> :

[Rapport Accelerate sur l'état du DevOps en 2014](#)

[Rapport Accelerate sur l'état du DevOps en 2015](#)

[Rapport Accelerate sur l'état du DevOps en 2016](#)

[Rapport Accelerate sur l'état du DevOps en 2017](#)

[Rapport Accelerate sur l'état du DevOps en 2018](#)

[Rapport Accelerate sur l'état du DevOps en 2019](#)

[Rapport Accelerate sur l'état du DevOps en 2021](#)

