

An aerial, high-angle photograph of a complex railway track system, likely in an urban environment. The tracks are made of steel rails on wooden sleepers and curve in various directions. In the upper left, there is a small, white, box-like control building with a staircase leading to it. In the upper right, a train is visible on one of the tracks. The background shows a multi-story building with many windows. The entire image has a blue tint.

Guide pratique de prise en main de DevOps

Table des matières

DevOps est-il fait pour vous ?	3
Développement et production sur de mauvaises voies	5
7 signes d'un processus dysfonctionnel	6
Conclusion : où débiter votre transformation DevOps ?	24



DevOps est-il fait pour vous ?

Dans l'économie des applications, les logiciels sont l'affaire de chaque entreprise. Dès lors, une approche DevOps devient rapidement l'une des disciplines les plus précieuses pour votre activité. Axée sur l'amélioration de la qualité des nouvelles applications et l'accélération de leur mise sur le marché, une méthodologie DevOps implique une intégration étroite entre développement et production.

Au départ perçu comme un simple effet de mode passager, le DevOps suscite désormais la réflexion au sein de nombreuses entreprises du monde entier.

Chacun se demande désormais si une telle approche ne serait pas pertinente pour sa propre entreprise.

Une organisation moyenne qui adopte DevOps constate une amélioration de 20 % de son délai de mise sur le marché, une progression de 22 % de la qualité de ses logiciels et une hausse de 17 % de la fréquence de ses déploiements d'applications, soit un accroissement de 22 % du nombre de clients et une hausse de 19 % de son chiffre d'affaires.¹

Que peut donc vous apporter DevOps ?

¹ Rapport TechInsights : « What Smart Businesses Know About DevOps », septembre 2013



Qu'est-ce que DevOps ?

Avant de savoir si nous pouvons utiliser DevOps, nous devons définir ce qu'il représente, mais aussi ce qu'il n'est pas.

DevOps n'est pas un produit ou une technologie spécifique. **DevOps est une méthodologie** qui réunit les fonctions souvent distinctes du développement de logiciels (Dev) et de la production et des opérations IT (Ops) en un seul processus intégré et continu.

DevOps permet de franchir les obstacles érigés entre le développement et la production. Il s'appuie sur les personnes, les processus et la technologie pour stimuler la collaboration et l'innovation tout au long du processus global de développement et de mise en production d'un logiciel. Les équipes de développement et de production IT doivent agir comme une seule et même entité, et se sentir comme tel.

Mais DevOps est infini. Comme un orchestre symphonique qui apprend une nouvelle partition ou une équipe sportive qui a atteint les qualifications, le développement et la production doivent continuer à progresser, en collaborant afin d'atteindre la perfection.

Développement et production sur de mauvaises voies

Le développement se concentre sur des innovations plus rapides et sur la mise en œuvre de nouveautés.

La production a quant à elle pour objectif la stabilité, le contrôle et la prévisibilité. Souvent, ces deux équipes ne rendent pas leurs comptes aux mêmes niveaux dans l'organisation. Elles semblent évoluer sur deux voies différentes. Peu importe la vitesse à laquelle elles avancent, elles ne se rencontrent jamais.

Seules, les équipes de développement et de production IT auront souvent du mal à communiquer, encore plus à collaborer, et resteront engluées dans des processus manuels. Résultat : des employés qui ne travaillent pas bien ensemble, des logiciels qui ne fonctionnent pas de manière fiable et des clients qui envisagent d'aller voir la concurrence.

Plus d'hésitation.



Section
suivante

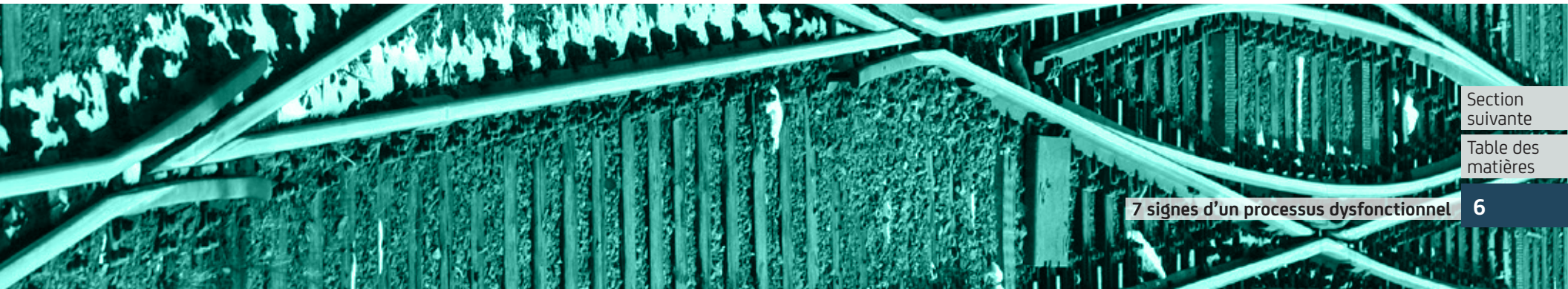
Table des
matières

7 signes d'un processus dysfonctionnel

Il peut être difficile d'engager les équipes de développement et de production vers un même objectif. Commencez donc par déterminer si vos équipes se trouvent effectivement sur des voies différentes. Les avertissements d'un processus dysfonctionnel peuvent notamment prendre les formes suivantes :

1. Vous détectez les défauts des logiciels à un stade avancé du cycle de vie ou, pire, en production.
2. Vous utilisez Agile pour accélérer le développement, mais tous les bénéfices s'évaporent lorsque l'application passe en production.
3. Vos développeurs et testeurs attendent sans cesse de pouvoir accéder aux ressources dont ils ont besoin, ce qui entraîne des retards.
4. Vous ne parvenez pas à identifier les problèmes au cours du développement, des tests et de la production.
5. Vous constatez que de simples erreurs humaines causent des ravages lors du développement et du déploiement.
6. Le développement considère que son travail est terminé lorsque l'application passe en production.
7. À chaque fois qu'un problème survient, tous se rejettent la balle.

Dans les pages qui suivent, nous allons examiner les signes les plus fréquents d'un processus dysfonctionnel, ainsi que certains conseils pratiques visant à dépasser le dysfonctionnement pour atteindre la pleine maturité DevOps.



Signe d'avertissement :

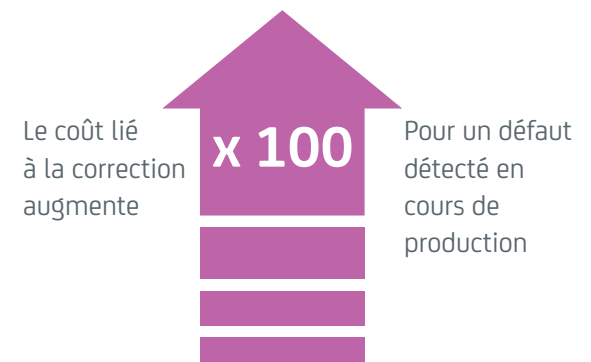
Les défauts des logiciels sont détectés à un stade avancé du cycle de vie

Il est impossible de créer des logiciels complexes sans aucun défaut. L'astuce consiste à détecter ces défauts au plus tôt dans le processus.



Avec l'approche classique en cascade du cycle du développement d'un logiciel, le coût lié à la correction d'un défaut augmente à chaque étape. Il peut être cent fois plus important si c'est un utilisateur final qui trouve un défaut majeur dans la production que si c'est vous qui le repérez au cours du développement.²

Malheureusement, avec la méthode en cascade, lorsque la planification du développement commence à se dérégler et qu'une échéance majeure se profile, les organisations manquent de temps pour les tests. Cette situation est propice aux erreurs et, du fait de la pression pour respecter les délais, l'application est envoyée en production, quel que soit le niveau de test atteint.



² « Software Defect Reduction Top 10 List », Computer, janvier 2001. <http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/82.78.pdf>

Comment revenir sur la bonne voie ?

En développant et testant en parallèle

Vous pouvez y parvenir en utilisant une nouvelle approche technologique appelée la virtualisation de services, qui peut imiter le comportement réel de l'environnement de production. Les ressources réelles d'infrastructures ne sont pas disponibles pour les développeurs et les testeurs. Cette approche recrée donc tout l'environnement dans un environnement virtuel. Les infrastructures, bases de données et serveurs back-end, ainsi que le reste de l'environnement, sont disponibles sous forme de service virtuel, de telle sorte qu'un grand nombre de personnes peuvent tester de multiples composants en même temps, sans se gêner les uns les autres et sans affecter l'environnement de production. Il devient ainsi possible d'assurer le développement et les tests en parallèle.

Mieux encore, il est possible d'automatiser ces tests, accélérant encore le processus.



Tests d'acceptation par l'utilisateur

Développement

Tests QA

Tests de performances

Production

Tests de régression

Tests d'intégration

Section
suivante

Table des
matières

Cette capacité d'imitation des performances de l'environnement de production gagne encore en réalisme avec l'exploration de données collaborative, pour laquelle les informations de performances intégrées dans l'environnement de production sont « explorées » pour découvrir des circonstances et schémas cachés.

Grâce à l'exploration de données, les développeurs et les ingénieurs qualité disposent de modèles de services virtuels qui se comportent encore davantage comme le monde réel, en offrant tous les avantages de la virtualisation de services grâce aux profils de performances imitant les conditions et scénarios métier rencontrés en production. Désormais, les développeurs et les ingénieurs des performances réduisent leurs contraintes grâce à un modèle toujours plus proche de la réalité, en émulant également les caractéristiques de comportement et de performances des véritables composants dans des situations réelles.

En détectant les problèmes plus tôt, vous créez des logiciels plus efficaces, plus rapidement et avec moins d'efforts.





Signe d'avertissement :

Les gains de temps permis par Agile s'évaporent lorsque l'application passe en production

La méthode Agile est une excellente approche pour accélérer le développement de logiciels. **Toutefois, la création d'un logiciel n'est qu'une étape de la mise à disposition d'une application pour les utilisateurs.**

Du fait du manque de collaboration entre les équipes de développement et de production, il est fréquent que les applications ne soient pas prêtes lorsqu'elles arrivent en production. Souvent, la production n'a pas été impliquée dans le développement, elle ne comprend donc pas vraiment comment déployer l'application. Il en résulte un grand nombre d'essais et d'erreurs lors du passage en production, soit une perte de temps.

Le développement seul ne peut pas mettre l'application entre les mains du client. Peu importe la rapidité et l'agilité avec lesquelles vous effectuez le processus de développement, **il est difficile d'accélérer la mise sur le marché si rien n'est fait pour accélérer le délai de mise en production.**

Section
suivante

Table des
matières

Comment revenir sur la bonne voie ?

Impliquez la production IT dès que possible

Agile est conçu autour d'un processus itératif qui répond rapidement aux besoins du marché. L'idée consiste à déployer des changements incrémentiels fréquents au lieu de stocker les changements en vue d'un grand lancement. La production fait tout simplement partie de ce besoin du marché.

Des ateliers Agile ne sauraient imaginer développer une application sans porter une attention toute particulière aux besoins du marché. Ils doivent apporter une attention similaire aux besoins de l'environnement de production. Le développement et les tests doivent également se dérouler dans un environnement aussi proche que possible de la vraie vie (prêt pour la production).

Une autre source potentielle de difficultés réside dans la réalisation des étapes en série. Même dans les ateliers Agile les plus efficaces, à certains moments, le travail est réalisé en grande partie par étapes. De tels processus en série peuvent ralentir le développement et le déploiement. **Les équipes doivent travailler en parallèle** autant que possible et, lorsqu'il est impossible de répartir les éléments pour des efforts parallèles, elles doivent chercher des moyens de planifier les étapes en amont dans le processus. Il n'y a pas de raison pour que la production attende que le développement ait terminé avant de commencer à travailler sur le déploiement de l'application ; impliquez la production dès que possible et planifiez le déploiement pendant le processus de développement.





Signe d'avertissement :

Les plannings sont affectés par le fait que les équipes doivent attendre des ressources

À chaque fois qu'une application complexe est créée, qu'il s'agisse d'un service front-end pour une application mobile ou d'un élément pour la couche d'intégration au cœur de l'architecture globale, elle utilise probablement des informations tirées d'autres systèmes et services dans toute l'entreprise.

Souvent, les développeurs et les testeurs n'ont pas un accès facile à ces services. Cela peut être dû au fait que les systèmes back-end, comme le système d'inventaire, sont disponibles pour un accès en test une fois par semaine seulement, pendant deux heures. Il se peut aussi qu'un système de production contienne des données sensibles qui ne peuvent pas être utilisées. Tous ces éléments constituent de sérieuses contraintes pour le processus de développement et de test d'une application.

Un jour, le directeur technologique d'une entreprise a résumé le problème comme suit :

« Je ne peux rien faire tant que je n'ai pas tout... or, je n'ai jamais tout ! »

Section
suivante

Table des
matières

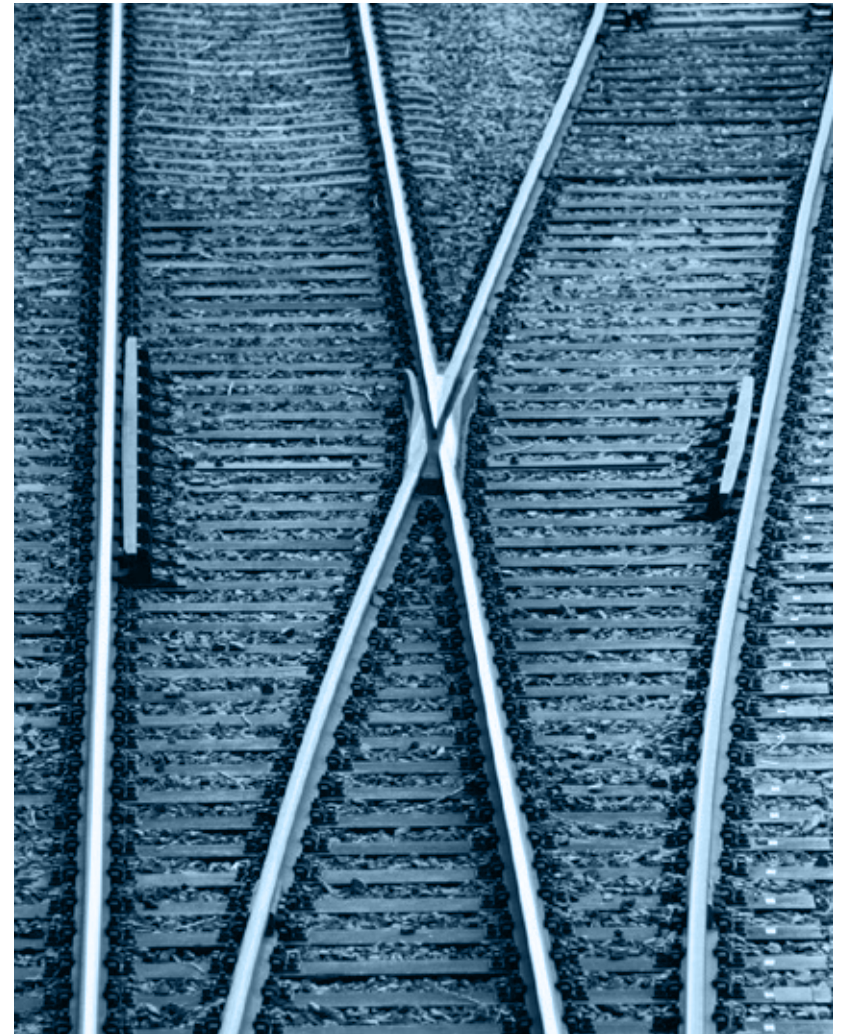
Comment revenir sur la bonne voie ?

La virtualisation des services peut aider à résoudre ce problème

Comme la virtualisation de services peut imiter le comportement réel de l'environnement de production, elle permet à un grand nombre de personnes de développer et tester de multiples composants en même temps, sans se gêner les uns les autres et sans affecter l'environnement de production.

Vous parvenez ainsi à éliminer les contraintes qui bloquent le développement et les tests d'applications dans de nombreuses organisations. Non seulement l'environnement de test est aujourd'hui un reflet beaucoup plus réaliste des conditions réelles, mais il permet également de tester simultanément différents composants. **Des étapes qui devaient auparavant être réalisées en série peuvent à présent être effectuées en parallèle.**

Le cycle complet de développement se décale vers la gauche tandis que les contraintes et les goulots d'étranglement sont éliminés.



Section
suivante

Table des
matières

Signe d'avertissement :

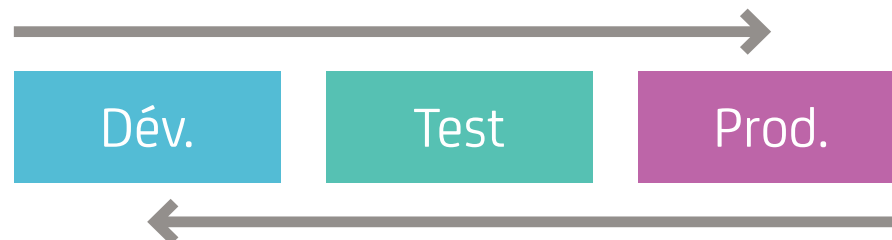
Les problèmes sont difficiles à repérer

Manque de collaboration entre les équipes de développement, de test et de production

Les équipes de développement, de test et de production travaillent souvent dans des environnements très différents et gèrent leur travail sur des systèmes distincts. Les défauts des logiciels peuvent dès lors être difficiles à corriger.

Résultats des tests non prédictifs

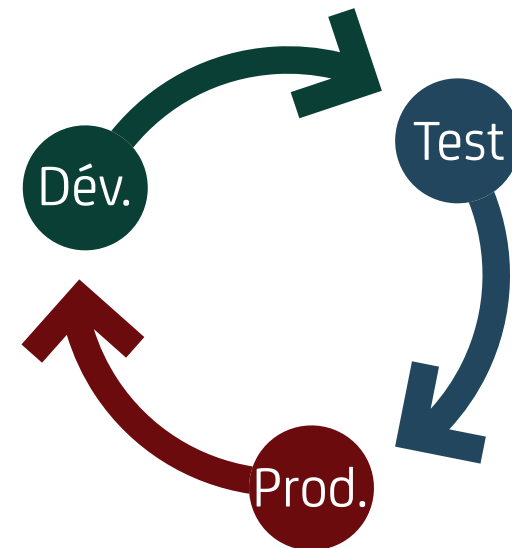
Aucune opportunité de test en conditions réelles



Aucun partage de retours d'expérience

Aucune communication entre

- Les systèmes
- Les processus
- Les personnes



Section
suivante

Table des
matières

Comme, souvent, les équipes de développement et les équipes de test ne disposent pas d'un accès fiable à l'environnement de production, elles se reposent sur des environnements de test qui, bien qu'ils offrent les fonctionnalités de l'environnement de production, ne présentent pas l'échelle de production complète. **Ce ne sont pas des environnements de test réalistes** qui reproduisent les conditions présentes en production. Par conséquent, il est fréquent que les applications défaillent lorsqu'elles passent en production parce que **les tests en amont ne prédisent pas le comportement de l'application dans le monde réel.**





Ensuite, comme la production, le développement et les tests utilisent tous des systèmes différents pour gérer leurs environnements, **la boucle de réaction**, qui doit être mise en place entre la production et le développement, **s'effondre**.

Le personnel de la production sait qu'il y a des problèmes avec l'application, mais si personne ne s'inquiète de signaler le problème à l'équipe de développement, celle-ci n'en saura jamais rien. Cette dernière ne pouvant pas superviser elle-même l'application en production, elle n'a qu'une vision secondaire des problèmes.

Comment revenir sur la bonne voie ?

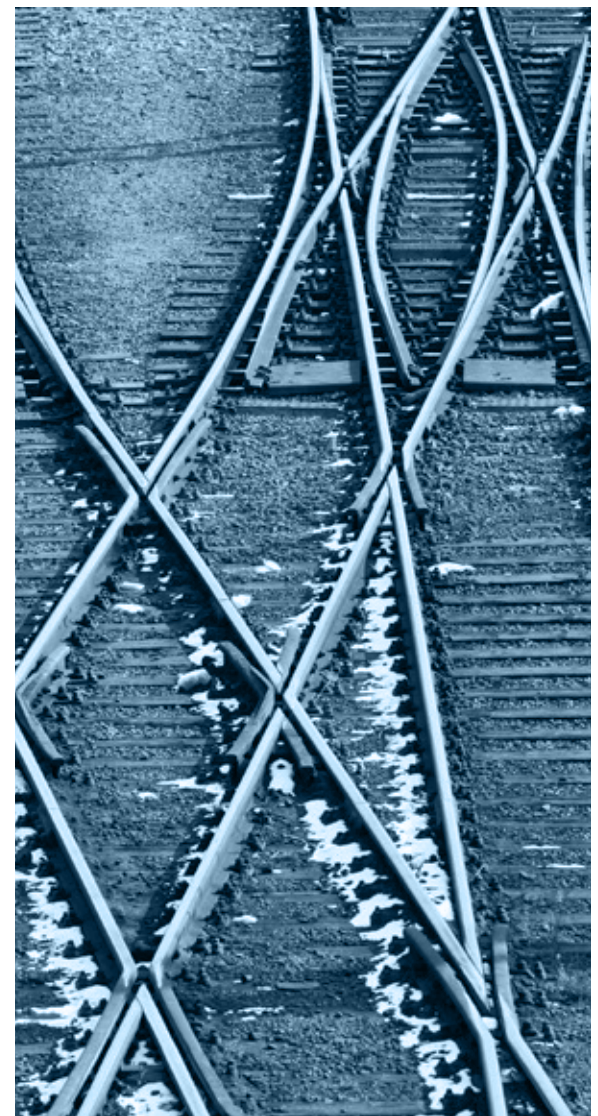
Il existe plusieurs manières de lutter contre ce dysfonctionnement

Tout d'abord, admettez que le développement et la production forment une seule équipe. Mélangez-les pour vous assurer que les **personnes** travaillent effectivement comme une équipe. Malgré ce changement d'organisation, établissez un **processus** pour recueillir les commentaires, puis pour suivre et partager les avancées liées à la résolution des problèmes.

Vous pouvez alors passer à l'étape suivante et utiliser la **technologie**. C'est très bien de changer les personnes et les processus, mais si vos systèmes fonctionnent à contre-courant, ce sera beaucoup plus difficile.

La technologie peut revêtir différentes formes. Premièrement, **assurez-vous que les systèmes** utilisés par le développement, les tests et la production pour gérer leur workflow (et les problèmes) sont **interopérables**, pour que le développement ait une visibilité sur les tests et la production, etc. Vous pouvez y parvenir en intégrant les systèmes existants, mais souvent, il est plus sensé de mettre en œuvre un système conçu spécifiquement pour créer un workflow intégré tout au long du processus de développement, de test et de mise en production.

Il est également judicieux de **mettre en place des environnements de test** qui reproduisent les conditions présentes dans l'environnement de production. Cela passe notamment par la virtualisation de services, appuyée par les scénarios de performances détaillés découverts en explorant les données contenues dans les journaux de l'environnement de production. Vous vous assurez ainsi que tous les tests reflètent **plus fidèlement les performances réelles de l'application**.



Section
suivante

Table des
matières



Signe d'avertissement :

Les erreurs humaines font des ravages et entraînent des pertes de temps pendant le déploiement

La mise en production de logiciels est un processus très largement manuel dans de nombreuses organisations. Seriez-vous surpris de découvrir que **la plupart des problèmes de logiciels sont dus à des erreurs de configuration** ? En fait, rien n'est mauvais dans l'application elle-même, mais le logiciel ne fonctionne pas parce qu'il n'est pas configuré correctement. Lorsque les configurations sont réalisées manuellement, il est inévitable qu'une erreur se glisse quelque part dans le processus.

L'erreur est humaine, mais elle peut tout gâcher. Les erreurs au cours du déploiement sont particulièrement préjudiciables parce qu'elles risquent d'être répliquées sur des milliers, voire des millions, de serveurs, d'ordinateurs de bureau et de dispositifs mobiles. **Les tests ne seront pas utiles, car la configuration nécessaire pendant les tests est différente** de celle requise en production.

Section
suivante

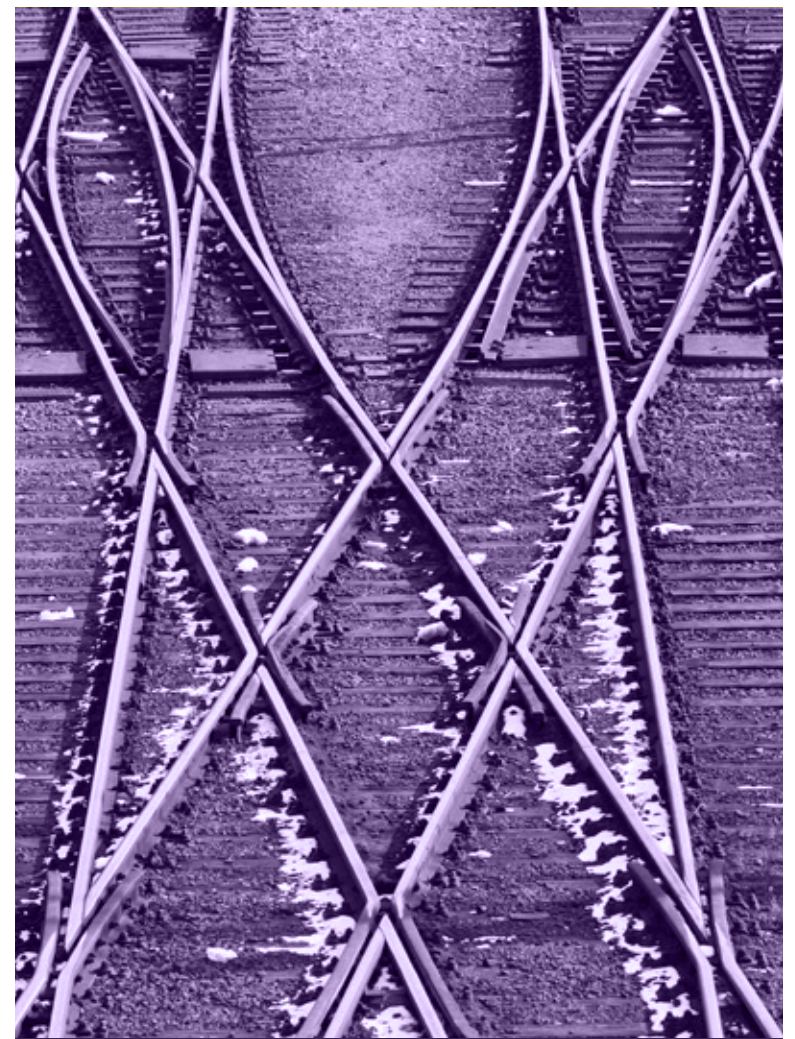
Table des
matières

Comment revenir sur la bonne voie ?

De nombreuses organisations ont tenté de résoudre ce problème en utilisant la génération de scripts. Bien que cette méthode soit certainement meilleure qu'un processus totalement manuel, elle génère de nombreux problèmes identiques à ceux d'un processus manuel. Du fait de toutes les « parties mobiles » et des différences de configuration entre les environnements, les scripts finissent par être aussi compliqués que les systèmes qu'ils tentent de déployer.

La génération de scripts ne suffit pas. Vous avez besoin de systèmes. Vous avez besoin **d'outils qui automatisent la mise en production d'un nouveau code et incluent la documentation** en interne, surtout dans les grandes équipes, afin d'appliquer réellement l'esprit de collaboration, d'intégration et de communication de DevOps. Grâce à cette automatisation de la mise en production, vous vous assurez qu'une configuration fonctionnant correctement lors des tests sera fidèlement traduite en un système qui fonctionnera en production. Elle élimine les erreurs humaines tout en accélérant l'ensemble du cycle de développement des logiciels. Ainsi, vous êtes mieux préparé pour une livraison continue, où les applications sont mises à jour toutes les quelques minutes, et non après plusieurs jours, semaines ou mois.

L'un des plus grands avantages de l'automatisation de la mise en production est le rétablissement, qui vous permet de revenir à un état correct connu si un problème surgit lors du déploiement. Avec les processus manuels, et même avec la génération de scripts, le rétablissement est un véritable cauchemar.





Signe d'avertissement :

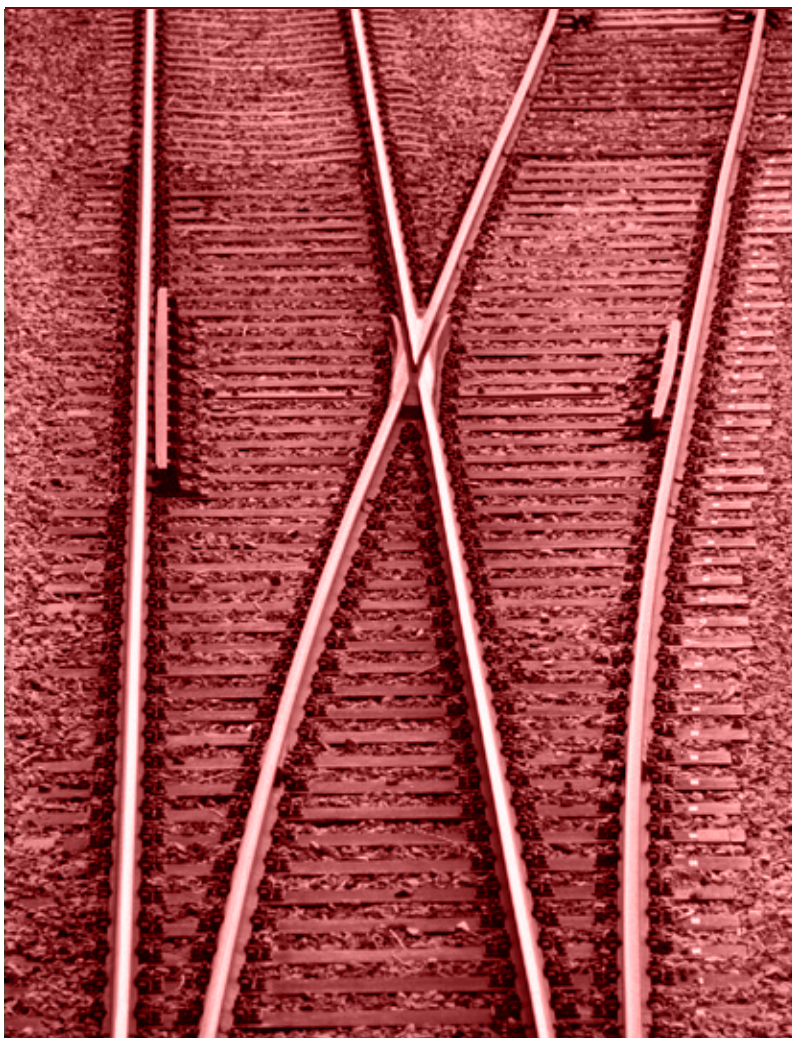
Le travail du développement est terminé lorsque l'application passe en production

Traditionnellement, les projets de développement d'applications exigent d'énormes efforts courant sur 18 à 24 mois pour la planification, le codage et les tests. Ensuite, l'application est envoyée en production. L'équipe de développement se réjouit, puis, tout le monde passe au projet suivant.

Si vous envisagez sérieusement d'utiliser DevOps, il n'existe pas de meilleur indicateur d'un processus dysfonctionnel que de voir l'équipe de développement jouer la balle et quitter le terrain.

Section
suivante

Table des
matières



Comment revenir sur la bonne voie ?

Une partie de la puissance de DevOps réside dans le fait qu'il vous permet de mettre vos applications en production sur une base quasi continue.

Au lieu de stocker une liste interminable de fonctions pour le grand lancement, le nouveau modèle doit proposer de nouvelles fonctions en continu. L'avènement des applications pour smartphone a instauré des attentes qui impliquent que les applications soient fréquemment actualisées. Si une application n'est pas mise à jour avec une ou deux nouvelles fonctions après quelques mois, les clients se demandent si cette application fait l'objet d'une maintenance correcte.

Avec la mise en production continue, l'équipe de développement termine une version et entame immédiatement la suivante, en ajoutant de nouvelles fonctions basées sur les exigences des utilisateurs et, dans l'idéal, guidées également par les commentaires de la production sur le niveau de performances de cette nouvelle version. Le développement et la production constituent véritablement un groupe intégré et collaboratif, à présent. La production sait ce qui est attendu du développement et le développement peut réellement envoyer son propre code en production, en utilisant des outils basés sur la charge avec des fonctions comme le rétablissement et le contrôle de version.

Signe d'avertissement :

À chaque fois qu'un problème survient, tous se rejetent la balle

En cas de problème, chacun commence-t-il par pointer l'autre du doigt et s'évertuer à démontrer que c'est la faute de l'autre ?

DevOps repose intégralement sur la collaboration et la coopération en équipe, et les doigts pointés sont délétères pour un environnement de travail collaboratif.

C'est un signe de dysfonctionnement majeur. Pour pouvoir mettre en œuvre DevOps, vous devez impérativement mettre fin à cette culture.



Section
suivante

Table des
matières

Comment revenir sur la bonne voie ?

La désignation d'un responsable est un comportement appris. Bien qu'il puisse être difficile à éradiquer, il existe des moyens de limiter ses effets.

Établissez clairement que la réussite ou l'échec émane d'un effort collectif. Il faut plusieurs personnes pour réussir, mais il faut également plusieurs personnes pour tout gâcher.

Définissez des objectifs communs pour tous les aspects du processus de développement, de test et de production. Ces objectifs doivent indiquer clairement que le développement ne peut pas être considéré comme une réussite tant que la production n'est pas également un succès, et vice versa. **Si tout le monde, équipe de développement et de production, a sa place depuis le début du cycle de développement d'un logiciel**, alors tout le monde est « embarqué » dans le processus et il est plus facile d'anticiper les problèmes et d'y remédier avant qu'ils ne glissent en production.

Considérez les erreurs comme un moyen d'apprendre. Prenez le temps d'analyser ce qui s'est mal passé, quelle est la leçon à tirer, ce qui doit être fait différemment et si c'est le symptôme d'un problème plus large qui doit être traité. Nombre de prétendues erreurs sont simplement la manifestation d'un processus imparfait. Améliorez le processus et éliminez les erreurs.

Automatisez les tâches pour éliminer autant d'erreurs potentielles que possible. Vous améliorerez ainsi la qualité du processus et vous permettrez aux gens de se concentrer sur les activités à valeur ajoutée, qui font réellement une différence pour l'organisation. Automatisez tout, puis utilisez l'analyse pour suivre et ajuster. Imaginez que Boeing construise des avions dépourvus de boîtes noires. Sans l'automatisation et l'analyse, leurs avions s'écraseraient sans laisser la possibilité de remédier au problème.



Conclusion :

Où débiter votre transformation DevOps ?

Admettre que vous avez un problème est la première étape permettant d'effectuer un changement significatif. En voyant la valeur que DevOps peut apporter à votre organisation, vous comprenez peut-être à quel point vous êtes entravé par des processus dysfonctionnels que vous ignorez.

Par conséquent, comment allez-vous sortir de ce dysfonctionnement et vous remettre sur la bonne voie DevOps ?

D'après notre travail de mise en œuvre de DevOps dans de très nombreuses organisations, il existe cinq actions qu'elles finissent toutes par entreprendre :

- Former des équipes chargées des applications qui intègrent toutes les disciplines (développement, test et production)
- Améliorer la formation, la communication et l'échange de compétences
- Réévaluer et reconstruire leur cycle de fourniture de services
- Évaluer les nouvelles technologies pour soutenir DevOps
- Choisir la bonne application ou la bonne ligne métier pour lancer DevOps



Peu importe vraiment dans quel ordre vous entreprenez ces actions ; l'important est de commencer.

Choisissez une application importante, qui suscite l'attention de tout le monde, comme point de départ de DevOps.

La tentation voudrait que vous commenciez par une « petite » application, mais d'après DevOps, vous retirez un bénéfice plus grand en vous confrontant à une application très importante et hautement visible, qui est déjà à l'origine de problèmes. Cela permet d'éliminer une grande partie de l'inertie interne face au changement et d'optimiser l'impact de la réussite de DevOps sur l'organisation. Si vous pouvez le faire sur un élément important, volumineux et peu engageant, vous pourrez réitérer cette réussite partout dans l'organisation.

Pour en savoir plus sur la manière dont vous pouvez bénéficier des avantages de la méthodologie DevOps pour votre organisation, visitez le site Web de CA Technologies.

<http://www.ca.com/fr/lpg/devops-portfolio.aspx>