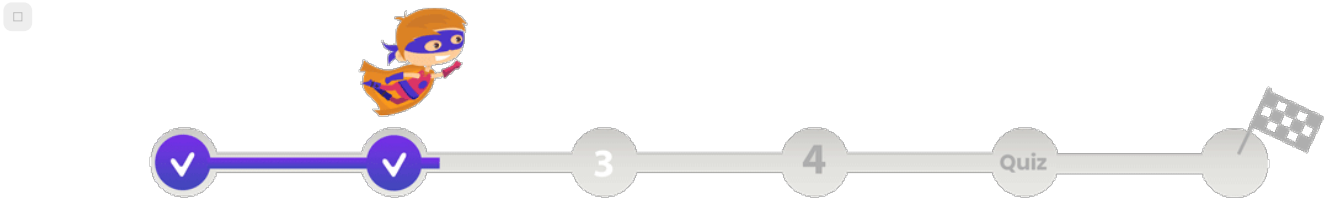


## 3 - Les avantages du DevOps



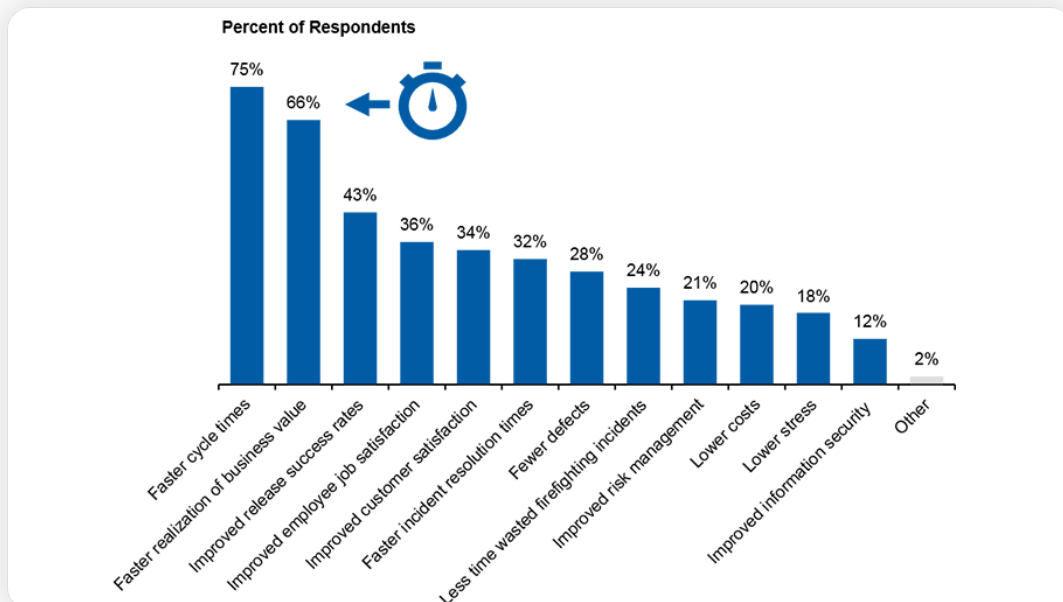
## 3. Les Avantages du DevOps

De nos jours, avec la multitude d'éditeurs logiciels qui foisonne sur le marché, il est important de pouvoir tirer son épingle du jeu. Il faut donc être très réactif, pouvoir proposer de nouvelles fonctionnalités le plus vite possible, mais il faut également prendre en compte au plus tôt les **remarques des clients**.



Selon le rapport « **2016 State of DevOps Report** » de chez Puppet, les **entreprises** qui ont **adopté la culture DevOps** annoncent en moyenne des **déploiements 200 fois plus fréquents** avec des **délais de mise en production 2 500 fois plus courts**.

Comme nous l'indique le graphique ci-dessous, il existe d'autres études qui arrivent aux mêmes conclusions, tout en indiquant d'autres avantages de la méthodologie DevOps.



### 3.1 Premier avantage : Réduire les risques et fluidifier les déploiements

Une parfaite cohésion au sein d'une équipe, une forte complicité entre les personnes, des employés qui se serrent les coudes lors des moments difficiles, de la confiance et une bonne dose de courage, voilà les principales conditions à la réussite d'un projet DevOps.

Avec ces ingrédients, vous pourrez réduire voire supprimer les risques qui restent trop souvent basés sur une mauvaise communication ou de l'incompréhension dans une équipe.

#### Exemple :

Bob est développeur pour une multinationale, il vient de transmettre les particularités du déploiement de son application à Alice qui est chargée du déploiement. **S'il y a des tensions** entre Bob et Alice il y a de **grandes chances** que le déploiement **échoue**, car Bob aura oublié de mentionner une particularité du déploiement. **Si au contraire il y a des liens forts** entre les deux employés, Bob pensera à **tous les points sensibles** du déploiement de son applicatif et viendra sûrement en **alerter** Alice en personne.

### 3.2 Deuxième avantage : Accélérer les rollback (retour en arrière)

C'est notamment la **mise en place d'outils d'automatisation** qui permet de **reproduire facilement** un problème, de le **corriger** et de **redéploier** une application rapidement sans impacter durablement la production. Des mécaniques de rollback vers la version précédente sont souvent fournies dans les outils d'automatisation ce qui permet de réduire d'autant plus le temps d'indisponibilité de la plateforme de production.



**Exemple :**

Bob vient de fournir son application conteneurisée à Alice pour le déploiement en production via la solution de dépôt de conteneur DockerHub. Alice demande à Eve, la chef du département qualité, si le conteneur de Bob a passé les tests de qualité. Eve lui donne son feu vert, Alice va donc **déployer** sur l'environnement de production **le même conteneur** que celui fourni par Bob et validé par Eve. En cas de problème non détecté auparavant par l'équipe d'Eve, Alice n'a qu'à remplacer le nouveau conteneur par l'ancien. Comme Alice utilise un orchestrateur de conteneur, **le rollback ne prendra que quelques secondes**.

### 3.3 Troisième avantage : Plus forte satisfaction client

Les deux points précédents influent forcément sur la **satisfaction client**. Si les déploiements en production sont stables, réguliers (proposant de nouvelle fonctionnalité) et prenant en compte leurs attentes, il n'y a pas de raison que les clients soient mécontents.



**Exemple :**

Alice utilise le logiciel "Software" de l'entreprise "Company" où travaille Bob. Elle découvre un **bug** lors de son utilisation quotidienne du logiciel. Elle **remonte le problème** via la plateforme de gestion de bug de "Company". Bob reçoit une notification, concernant le ticket d'Alice, il **reproduit le bug** et lui donne une **criticité** puis le rajoute dans le **backlog**. Parallèlement, Alice est **remerciée** par Bob pour son retour sur l'application, et lui indique que le bug sera traité **dans les plus brefs délais**. Le bug est **corrigé** lors du prochain sprint et est poussé en production automatiquement dans la foulée. Le **service qualité valide** que le bug a bien été corrigé en production. Alice est alors notifiée que le bug est corrigé.



Pour voir la fiche complète et les documents attachés, rendez-vous sur <https://elearning.26academy.com/course/play/5aa2649f53e5ad694aa0e5fc>