

Logiciels : comment réduire l'impact des défauts en production ?

Malgré les efforts déployés pour les détecter et les corriger en amont, les défauts logiciels continuent de se propager en production. Mais il est possible d'en limiter l'impact. C'est le principal enseignement d'une [enquête](#)* américaine publiée par le groupe IT Micro Focus.

Pour 73% des développeurs et dirigeants de grandes entreprises interrogés, les conséquences d'erreurs de code passées en production sont plus « gérables » et moins coûteuses dans un environnement DevOps que traditionnel.

Ils sont plus nombreux encore (83%) à considérer que corriger les défauts applicatifs prend moins de temps lorsque des méthodes agiles sont utilisées par l'organisation concernée.

Mais le DevOps n'est pas une combinaison miracle.

DevOps et sécurité

Quand une approche Agile/DevOps est adoptée, 40% des défauts logiciels en moyenne ont un « impact faible » sur les utilisateurs, l'activité et les revenus de l'entreprise. 33% de ces vulnérabilités auraient un « impact moyen » et 27% un « impact élevé ». Ces taux sont respectivement de 38%, 32% et 30% avec une approche traditionnelle du développement.

En conséquence, l'agilité et le DevOps n'empêchent pas le passage en production de défauts logiciels. Mais ces derniers sont plus susceptibles d'avoir un faible impact.

Par ailleurs, globalement, 84% des répondants pensent que les défauts logiciels « à faible impact » n'entament pas la réputation d'une marque. Mais ils redoutent une accumulation qui exposerait davantage le système d'information aux pannes et aux pertes de données.

En outre, devant les erreurs fonctionnelles et le déficit de performance, les défauts [de sécurité](#) sont ceux qui impactent le plus le chiffre d'affaires et la réputation de la marque.

Micro Focus recommande donc aux équipes en charge des développements applicatifs de :

- corriger les défauts connus à faible impact avant la publication du programme.
- opter pour des méthodes Agiles/DevOps qui permettront d'en réduire l'impact.
- tester plus tôt pour identifier et corriger plus rapidement, et limiter les coûts.
- donner la priorité aux tests de sécurité.

Car la maintenance corrective coûte cher.

*L'enquête a été menée en ligne par Harris Poll pour Micro Focus auprès de 200 décideurs et développeurs de grandes entreprises.