



# L'IA accélère le développement logiciel. Mais elle accélère aussi les bugs.

Pourquoi le QA devient plus critique que jamais — et ce que les équipes engineering doivent faire dès maintenant.

QUOREXIS · SOFTWARE TESTING & QUALITY ENGINEERING





# Les outils d'IA transforment le développement

Les modèles de génération de code comme GitHub Copilot, Cursor ou GPT-4 ont fondamentalement changé la façon dont les équipes produisent du logiciel. Ce qui prenait des jours se réalise en heures. Ce qui prenait des semaines se réalise en jours.

## Génération de code

L'IA produit des centaines de lignes en quelques secondes, réduisant le temps de développement de 30 à 50 %.

## Livraison accélérée

Les fonctionnalités atteignent la production plus vite, compressant les cycles sprint et les délais de mise sur le marché.

## Cycles de release courts

Les équipes déploient plusieurs fois par jour, là où elles déployaient une fois par semaine il y a encore deux ans.



# La vitesse crée de nouveaux risques

Plus de code généré signifie plus de surface d'erreur. L'IA ne comprend pas le contexte métier, les contraintes d'architecture ni les effets de bord en production. Elle produit du code plausible — pas nécessairement correct.

## Plus de code

Le volume de code produit explose, rendant la revue manuelle exhaustive impossible. Les équipes QA sont dépassées si elles ne s'adaptent pas.

## Plus de complexité

Le code généré par l'IA peut introduire des patterns non standards, difficiles à maintenir, à déboguer et à tester de façon systématique.

## Plus de dépendances

Les suggestions de l'IA tirent souvent des bibliothèques externes supplémentaires, augmentant la surface d'attaque et les risques de compatibilité.

# On mesure la vitesse. Rarement la qualité.

## Ce que la plupart des équipes mesurent



### Vélocité de sprint

Nombre de story points livrés par itération.



### Fréquence de déploiement

Nombre de releases par semaine ou par mois.



### Lead time

Délai entre la demande et la mise en production.

## Ce que très peu d'équipes mesurent

### → Taux de défauts en production

Combien de bugs atteignent les utilisateurs réels.

### → Couverture de tests effective

Quelle proportion du code est réellement testée.

### → Coût de la non-qualité

Impact financier des incidents et régressions.

❏ Les entreprises qui ne mesurent que la vitesse optimisent pour livrer — pas pour fiabiliser.

# Le QA n'est plus seulement du testing

# Il devient une couche de gouvernance pour la livraison logicielle.

Dans un monde où l'IA génère du code à grande vitesse, le Quality Assurance évolue vers un rôle stratégique. Il ne s'agit plus de chercher des bugs après le développement — il s'agit d'intégrer des garde-fous structurels tout au long du cycle de vie du logiciel.



## Gouvernance de la qualité

Définir des standards, des seuils de qualité et des politiques de validation qui s'appliquent à tout le code, qu'il soit écrit par un humain ou généré par l'IA.



## Intégration continue

Le QA s'intègre dans chaque étape du pipeline CI/CD, automatisant les vérifications pour maintenir le rythme de livraison sans sacrifier la fiabilité.



## Observabilité de la qualité

Créer une visibilité complète sur l'état de santé du logiciel à travers des tableaux de bord, des métriques et des alertes en temps réel.

# Les indicateurs QA essentiels à suivre

Un programme QA mature se mesure. Voici les quatre dimensions clés que toute équipe engineering sérieuse doit instrumenter et monitorer en continu.

1

## Couverture de tests

Pourcentage du code couvert par des tests automatisés. L'objectif minimal recommandé est de 80 % pour les composants critiques.

2

## Taux de réussite des tests

Proportion de tests qui passent à chaque build. Un taux dégradé signal une instabilité systémique avant qu'elle n'atteigne la production.

3

## Defect leakage

Taux de défauts qui échappent aux phases de test et atteignent les utilisateurs finaux. L'indicateur le plus direct de la maturité QA.

4

## Indicateurs de performance

Temps de réponse, disponibilité, charge — la qualité ne se limite pas à l'absence de bugs mais englobe l'expérience utilisateur globale.

# L'IA va continuer d'accélérer

## La promesse de l'IA

Les équipes qui adoptent l'IA dans leur workflow de développement gagnent un avantage concurrentiel réel : plus de fonctionnalités, plus vite, avec moins d'effectifs. Cette tendance est irréversible.

### Productivité x2 à x5

Les développeurs augmentés par l'IA livrent significativement plus de valeur par sprint.

### Time-to-market réduit

Les cycles de développement s'accélèrent, permettant d'itérer plus vite sur le produit.

## La réalité sans QA discipliné

La fiabilité d'un système ne se résume pas à la vitesse à laquelle il a été construit. Sans discipline de testing, l'accélération devient un multiplicateur de risques — pas de valeur.

### Dettes techniques explosives

Le code non testé s'accumule jusqu'à rendre les évolutions impossibles ou trop coûteuses.

### Incidents en production

Les bugs critiques en production érodent la confiance des utilisateurs et le chiffre d'affaires.

📌 **La fiabilité dépendra toujours de la discipline de testing.** L'IA ne remplace pas la rigueur — elle en amplifie l'importance.



# Un logiciel fiable nécessite de la structure

La qualité n'est pas une étape à la fin du développement. C'est une discipline qui s'intègre à chaque ligne de code, chaque release, chaque décision d'architecture.

## Quorexis · Software Testing & Quality Engineering

Nous accompagnons les CTOs, Heads of Engineering et équipes produit dans la mise en place de stratégies QA adaptées à l'ère de l'IA — des frameworks de tests automatisés aux programmes de qualité à l'échelle de l'organisation.

**Test. Trust. Deliver.**

Test Automation

QA Strategy

Quality Engineering