

# Quelles sont les erreurs fréquentes dans les projets sans tests ?



# #1 : Bugs détectés tardivement (ou jamais)

Sans stratégie de test, les bugs restent dans le produit et sont souvent découverts trop tard, en production.

## Conséquences :

- Coûts de correction multipliés (jusqu'à 100x par rapport à une détection en phase de conception).
- Impact sur la réputation et la satisfaction client.

Exemple :

Une application bancaire livrée sans tests exhaustifs affiche des montants erronés, entraînant une perte de confiance massive chez l'utilisateurs



## #2 : Régressions non maîtrisées

Sans tests de régression automatisés, chaque modification ou nouvelle fonctionnalité risque de casser des parties existantes.

### Conséquences :

- Dépendance à des corrections rapides, parfois dangereuses (hotfixes).
- Des sprints tournés vers la réparation plutôt que l'innovation.



### Exemple :

Lors d'une mise à jour, une boutique en ligne voit son système de paiement cesser de fonctionner.

Résultat : perte de ventes et panique des équipes.



## #3 : Mauvaise adéquation avec les besoins utilisateurs

Sans tests d'acceptation ou vérification continue avec les parties prenantes, le produit final risque de ne pas correspondre aux attentes.

### Conséquences :

- Fonctionnalités inutiles ou mal implémentées.
- Retours négatifs des utilisateurs, nécessitant des refontes coûteuses.

Exemple :

Un logiciel de gestion de projet livré avec des fonctionnalités avancées mais inutilisables car l'interface est trop complexe.



## #4 : Surcoûts liés à l'absence d'automatisation

Les tests manuels non systématisés prennent du temps et ne couvrent pas tous les scénarios critiques.

### Conséquences :

- Coût élevé en temps humain.
- Faible couverture des cas de test (souvent <50 %).



### Exemple :

Un projet où chaque cycle de test manuel prend des jours, ralentissant les déploiements et frustrant les équipes.

## #5 : Communication et collaboration faibles

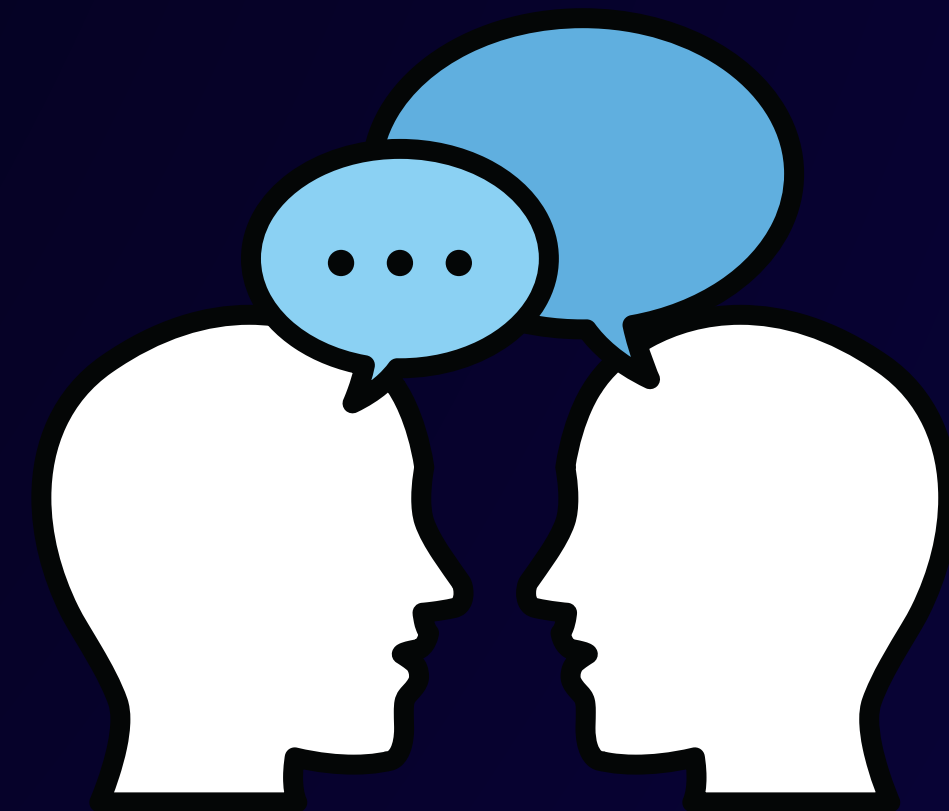
Sans stratégie de test partagée, les équipes ne savent pas ce qui a été validé ou non, ce qui génère des malentendus.

### Conséquences :

- Des zones critiques oubliées.
- Des conflits entre développeurs, testeurs et parties prenantes.

Exemple :

Un projet agile où les tests ne sont pas alignés sur les user stories, menant à des décalages entre la vision produit et la livraison.





## #6 : Manque de scalabilité et de robustesse

Sans tests de performance ou de charge, le produit peut ne pas supporter une montée en charge.

### Conséquences :

- Pannes lors d'un pic d'utilisateurs.
- Perte de clients et mauvaise image publique.

Exemple :

Une application mobile incapable de gérer l'afflux d'utilisateurs lors de son lancement viral.



# Comment éviter ces erreurs ?

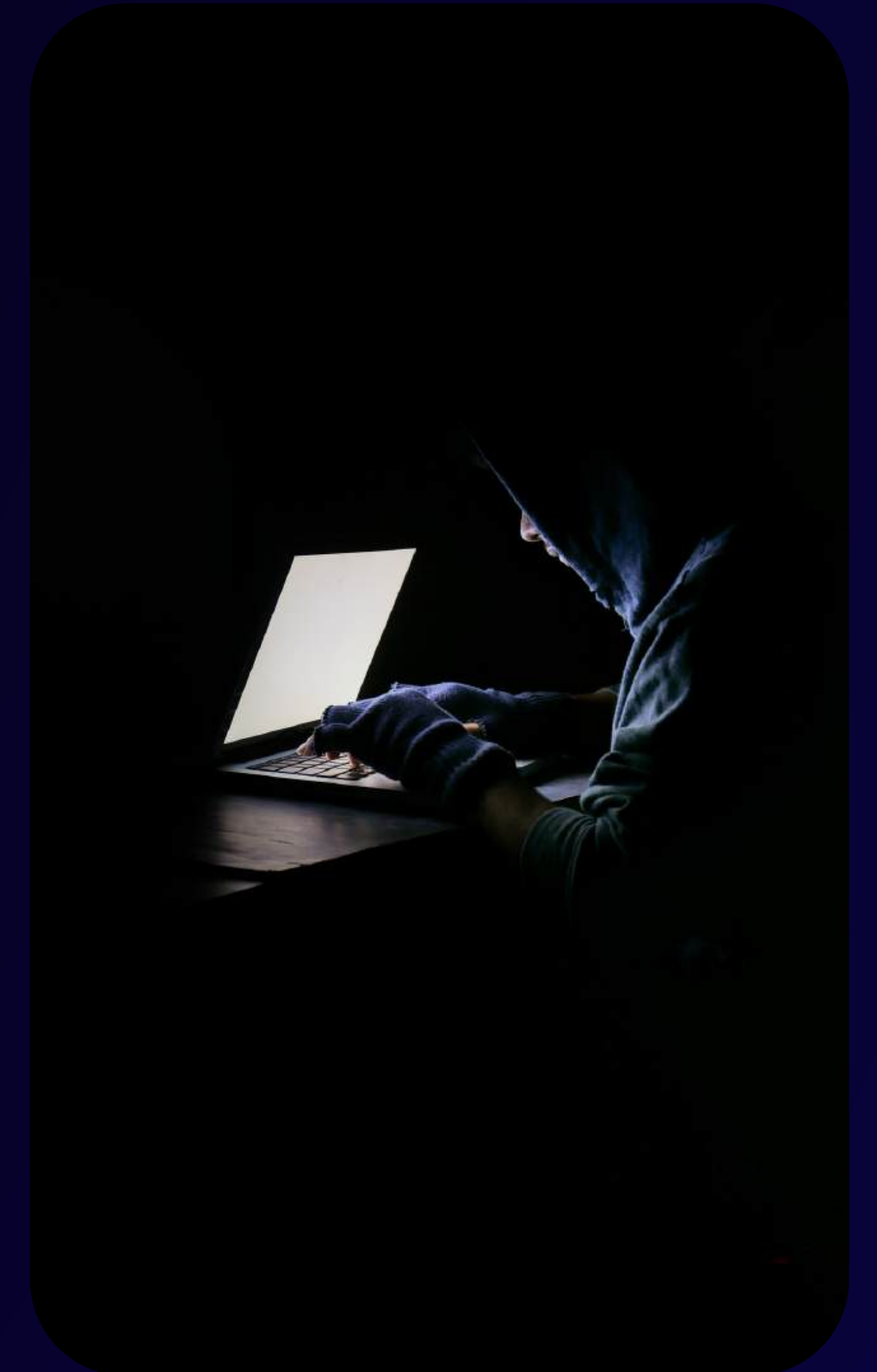
- Adoptez une stratégie de test claire dès le début : Intégrez des tests fonctionnels, d'acceptation et de régression.
- Investissez dans l'automatisation : Utilisez des outils comme d'automatisation de tests pour couvrir plus de scénarios en moins de temps.
- Impliquez toutes les parties prenantes : Les tests ne concernent pas seulement les testeurs, mais aussi les développeurs et les Product Owners.
- Mettez en place des métriques : Suivez la couverture des tests, le taux de réussite et la fréquence des bugs pour évaluer vos efforts.
- Privilégiez les tests continus : Adoptez une approche CI/CD pour garantir des feedbacks rapides et une qualité constante.



# Conclusion

Les tests, un investissement rentable

Ne pas tester correctement, c'est jouer avec le temps, l'argent et la confiance des clients. Les tests adéquats ne sont pas un coût, mais une garantie pour un produit fiable, performant et adapté à son marché.



# CENTRAL TEST AGENCY

QUALITY ASSURANCE - USER EXPERIENCE