

## Les types de bugs et leur gravité

Dans un projet informatique, tous les bugs ne se valent pas. Certains bloquent complètement le travail, d'autres sont gênants, et certains n'ont qu'un impact mineur. Classer les bugs par **niveau de gravité** est essentiel pour **prioriser les corrections**, organiser le travail de l'équipe et respecter les délais.

5TTR

6TTR



Découverte

## Pourquoi classer les bugs

Sans classification, tout bug semble urgent. En réalité, corriger un bug graphique mineur avant un bug bloquant est une **mauvaise décision de gestion de projet**.

La classification permet de :

- décider **quoi corriger en premier**,
- communiquer clairement dans l'équipe,
- éviter les débats subjectifs ("ce bug est grave / pas grave").

## Bug bloquant (Blocking / Critical)

Un bug bloquant empêche totalement l'utilisation du logiciel ou l'avancement du projet.

Caractéristiques :

- l'application ne démarre pas,
- une fonctionnalité essentielle ne fonctionne pas du tout,
- perte ou corruption de données,
- impossibilité de livrer ou de tester.

Exemples :

- crash au lancement,
- impossible de se connecter,
- paiement impossible alors que c'est la fonctionnalité centrale.

👉 **Priorité absolue.** Un bug bloquant doit être corrigé immédiatement.

# Bug majeur (Major)

---

Un bug majeur affecte fortement le fonctionnement, mais **une solution de contournement existe**.

Caractéristiques :

- fonctionnalité importante partiellement cassée,
- comportement incorrect mais non bloquant,
- impact fort sur l'expérience utilisateur.

Exemples :

- un formulaire fonctionne, mais avec des erreurs fréquentes,
- une fonctionnalité clé est lente ou instable,
- affichage incorrect qui gêne l'utilisation.

👉 **À corriger rapidement**, juste après les bugs bloquants.

# Bug mineur (Minor)

---

Un bug mineur a un impact limité sur l'utilisation du logiciel.

Caractéristiques :

- le logiciel reste utilisable,
- l'erreur est surtout esthétique ou secondaire,
- aucun impact sur les données ou la logique principale.

Exemples :

- texte mal aligné,
- faute d'orthographe,
- couleur incorrecte sur un bouton.

👉 **À corriger si le temps le permet**.

# Bug trivial (Cosmetic / Trivial)

---

Ce type de bug n'a quasiment **aucun impact fonctionnel**.

Caractéristiques :

- purement visuel,
- très faible gêne pour l'utilisateur,
- n'affecte ni la compréhension ni l'usage.

Exemples :

- icône légèrement décalée,
- animation imparfaite,
- détail graphique mineur.

👉 Souvent ignoré ou corrigé en fin de projet.

## Autres classifications courantes

---

Selon les équipes, on peut aussi rencontrer :

- **Bug de performance** : lenteur, consommation excessive de ressources.
- **Bug de sécurité** : faille, accès non autorisé, fuite de données.
- **Bug de compatibilité** : problème selon le navigateur, l'OS ou le matériel.
- **Bug de logique** : le programme fonctionne, mais pas comme attendu.

Ces catégories décrivent la **nature du bug**, tandis que *bloquant / majeur / mineur* décrit sa **gravité**.

## Gravité vs priorité

---

Attention :

- **Gravité** = impact réel du bug.
- **Priorité** = ordre dans lequel on décide de le corriger.

Un bug mineur peut devenir prioritaire s'il touche beaucoup d'utilisateurs ou s'il est très visible.

## Synthèse

---

Ce qu'il faut retenir :

- Tous les bugs n'ont pas la même gravité.
- **Bloquant** : empêche de travailler ou d'utiliser le logiciel.
- **Majeur** : gêne fortement, mais sans blocage total.
- **Mineur** : impact limité.
- **Trivial** : impact négligeable.
- Classer les bugs permet de **corriger efficacement** et de **tenir les délais**.

Bien gérer les bugs, ce n'est pas tout corriger tout de suite, c'est **corriger ce qui compte vraiment en premier**.