Engineer

Qualité du code

aptitude du code à bien fonctionner et à évoluer

Ce que c’est

La qualité du code tient à la manière dont le code source réalise le comportement attendu et à la facilité avec laquelle on peut le comprendre, le vérifier et le faire évoluer. Prenons une petite fonction de facturation. Une version obtient le bon montant pour la commande de démonstration, parce qu’elle suit précisément le chemin prévu par son auteur. Une autre indique la devise, exprime la règle d’arrondi à un seul endroit et traite les avoirs de façon explicite. Les deux peuvent passer le test du cas courant. Pour enquêter sur une facture contestée, la seconde offre un meilleur point de départ.

L’exactitude se joue aussi aux limites, par exemple en cas de données invalides, de nouvelle tentative ou d’opération effectuée à moitié. La lisibilité se mesure à ce que la personne suivante peut comprendre des décisions prises. La facilité d’évolution dépend de l’étendue du prochain changement et de la possibilité d’en vérifier les effets. Une fonction courte peut calculer une taxe incorrecte. Une fonction complexe peut être correcte aujourd’hui et rendre très difficile l’ajout d’une devise demain.

La revue de code contribue à apprécier ces propriétés. Un contrôle bloquant empêche les changements d’avancer quand certaines vérifications échouent. La dette technique désigne les coûts futurs qu’une décision entraîne. La qualité du code est l’état du code lui-même. La règle de revue de Google traduit cette idée en une décision quotidienne, en demandant si chaque modification améliore la santé de l’ensemble du code.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que la qualité du code ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Une seule équipe transforme une idée en un produit auquel vos clients peuvent se connecter, sans le faire passer entre trois prestataires.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Application GIS : revenus multipliés par 10 — 10x croissance du chiffre d'affaires portée par le produit cartographique

Application GIS : revenus multipliés par 10 Contactez-nous →

Pourquoi c’est important

C’est généralement une nouvelle exigence qui révèle la qualité du code. Un service produit un devis, émet une facture puis établit éventuellement un avoir. Si chaque opération recalcule indépendamment la même remise, faire évoluer les conditions d’éligibilité oblige à trouver toutes les copies. Une divergence peut seulement se remarquer au moment du premier remboursement. Si une règle commune est exprimée à un endroit connu et que chaque opération lui fournit ses données clairement, la personne qui relit le changement peut en prévoir la portée.

Une preuve utile correspond à un risque. Pour les montants, on choisira un exemple où arrondir les lignes donne un autre résultat qu’arrondir le total. Pour une action susceptible d’être relancée, on vérifiera si la deuxième requête déclenche un second débit. Lorsqu’un calcul est exécuté à chaque commande, mieux vaut mesurer le délai sous la charge attendue avant d’ajouter un cache. Chaque observation permet une conclusion dans un périmètre précis.

Il y a aussi un enjeu de transmission. Les personnes changent de projet et la personne chargée de la revue peut découvrir le module. Des noms précis, un exemple exécutable et une explication courte d’une contrainte inhabituelle permettent de retrouver et de vérifier une décision sans demander de l’aide à son auteur. Cette autonomie compte pendant une panne autant que pour intégrer quelqu’un à une équipe.

À quoi cela ressemble en pratique

Imaginons qu’une règle demande d’arrondir la taxe sur le total par taux d’imposition. Sur une facture à deux lignes, la taxe brute de chacune vaut 0,005 unité monétaire. Si on arrondit chaque ligne au centime en traitant les demis à la

hausse, on obtient 0,02 au total. Si on additionne avant d’arrondir une seule fois, on obtient 0,01. On peut refaire ce calcul à la main. L’exemple illustre une décision qui, pour une vraie facture, relève des exigences du produit et des règles applicables.

Le contrat de la fonction peut préciser l’unité attendue et l’ordre des opérations. Il faut garder la précision requise pendant les calculs intermédiaires. Le regroupement doit rester visible et l’arrondi intervenir au point convenu. Un test avec les chiffres précédents distingue bien les deux méthodes. Un autre peut mêler deux taux pour éviter qu’un remaniement ultérieur remplace les groupes par une somme globale. Si les avoirs obéissent à une règle différente, ajoutons un exemple propre à cette règle.

À la revue, suivons le montant affiché, celui qu’on exporte et celui qui sert lors du remboursement. Les anciennes factures gardent les montants enregistrés lors de leur émission. Un essai de la chaîne complète vérifie que les appels emploient les bonnes unités. Un test isolé dit si le calcul fonctionne avec ses entrées, tandis que l’essai de bout en bout indique si les bonnes entrées lui parviennent réellement.

Notre note sur la vente du dernier billet une seule fois applique la même démarche à un autre domaine. Elle pose une égalité que les stocks doivent toujours respecter, prévoit des tests concurrents et propose de surveiller cette égalité en fonctionnement. Une personne qui relit une modification peut ainsi repérer le résultat que plusieurs parcours doivent préserver.

Où les problèmes surviennent

Un indicateur peut être pris pour une garantie. La couverture montre quelles lignes ou branches les tests ont parcourues. Si le résultat attendu traduit la mauvaise règle d’arrondi, le test peut réussir. Un indice de complexité aide à repérer du code à lire. Remplacer une comparaison claire par des appels à plusieurs fonctions peu explicites peut pourtant rendre un incident plus difficile à comprendre. Avant de poser un seuil, identifions les défauts qu’il doit permettre de découvrir et examinons des exemples de part et d’autre.

On peut aussi consacrer trop de temps à un nettoyage peu important. Une fonction mal nommée dans un rapport occasionnel n’a pas la même portée qu’une condition floue qui s’applique à chaque prélèvement. Indiquons le risque

constaté, les corrections requises et les améliorations qui peuvent attendre. Une préférence de présentation se règle souvent par une convention et un outil de formatage.

Enfin, un changement local peut déplacer un problème ailleurs. Mutualiser les remises a du sens lorsque plusieurs opérations suivent la même règle commerciale. Si un remboursement et une promotion suivent deux politiques différentes, une fonction qui devine laquelle s'applique sera difficile à relire. Commençons par préciser quels éléments coïncident. Partageons la partie stable et rendons les différences visibles. Une autre personne doit pouvoir expliquer la règle à partir du code et en vérifier le résultat.

Pour aller plus loin

- Règle de décision de Google pour la revue de code
- Ce que Google vérifie lors d'une revue
- Tests unitaires dans Software Engineering at Google

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que la qualité du code ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/code-quality/