Engineer

Якість коду

наскільки код коректний, зрозумілий і придатний до змін

Що це таке

Якість коду показує, наскільки добре вихідний код реалізує очікувану поведінку і наскільки легко його зрозуміти, перевірити та змінити. Уявімо функцію виставлення рахунків. Одна версія повертає правильну суму для демонстраційного замовлення, бо повторює саме той порядок дій, який розглядав розробник. Інша явно зазначає валюту, тримає правило округлення в одному місці й окремо враховує повернення коштів. Обидві можуть пройти звичайний тест. Друга дає зрозумілу відправну точку, якщо клієнт оскаржить суму.

Коректність стосується і граничних випадків, зокрема хибних вхідних даних, повторного запиту та частково виконаної операції. Зрозумілість залежить від того, чи можна відновити ключові рішення, читаючи код. Придатність до змін визначає, наскільки далеко поширюється вплив нової вимоги і чи можна перевірити наслідки. Коротка функція може рахувати податок неправильно. Заплутана може видавати правильну суму тепер, але ускладнити перевірку нової валюти згодом.

Рецензування коду допомагає оцінити ці властивості для конкретної зміни. Шлюз якості може зупинити роботу, якщо вибрані перевірки не пройдено. Технічний борг описує майбутні витрати, зумовлені певним рішенням. Якість коду стосується стану самого коду. Правило Google пропонує оцінювати кожну зміну за тим, чи поліпшує вона стан кодової бази в цілому.

Застосуйте поняття до рішення

Застосуйте «Що таке якість коду?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Одна команда веде ідею від ескізу до продукту, у який ваші клієнти вже можуть увійти, а не перекидає роботу між трьома підрядниками.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Збільшила дохід GIS-застосунку в 10 разів — 10x зростання виторгу завдяки картографічному продукту

Збільшила дохід GIS-застосунку в 10 разів Зв'язатися з нами →

Чому це важливо

Якість особливо помітна, коли змінюється вимога. Сервіс готує пропозиції, виставляє рахунки й повертає кошти. Якщо та сама знижка обчислюється окремо в кожному з трьох шляхів, наступна акція вимагає знайти й оновити всі копії. Розбіжність може стати помітною тільки після повернення першого замовлення. Спільне правило у відомому місці та явно передані дані дають змогу зрозуміти обсяг зміни і вибрати потрібні перевірки.

Обирайте докази відповідно до ризику. Для грошових сум корисний випадок, у якому округлення окремих рядків дає інший підсумок, ніж округлення суми. Для повторюваних операцій важливо перевірити, чи другий запит не спричиняє другого списання. Якщо обчислення запускається під час кожної покупки, перед додаванням кешу варто виміряти час відповіді під очікуваним навантаженням. Спостереження дає відповідь у межах конкретних умов.

Якість також допомагає передавати роботу. Учасники змінюють проєкти, тож той, хто читає наступну зміну, може не знати історії модуля. Влучні назви, приклад, який можна запустити, і коротке пояснення незвичної вимоги дають змогу самостійно перевірити рішення. Це важливо і для навчання, і для чергового, який розбирається зі збоєм.

Як це виглядає на практиці

Припустімо, що податок округлюють після підсумовування за кожною ставкою. У рахунку два рядки, податок до округлення становить по 0,005 грошової одиниці. Якщо округлити кожний рядок до сотих із правилом округлення половини вгору, сума становитиме 0,02. Якщо спочатку додати значення, а потім округлити один раз, результат буде 0,01. Це можна перевірити вручну. Правило для справжнього рахунку має спиратися на вимоги продукту й чинні норми.

У контракті функції варто назвати одиниці вхідних значень і порядок дій. Проміжні результати мають зберігати потрібну точність. Групування має бути помітним, а округлення відбуватися на погодженому кроці. Тест із

наведеними числами відрізняє два способи. Інший тест із різними ставками допомагає виявити, що хтось замінив групування однією загальною сумою. Для окремої політики повернення потрібен окремий приклад.

Під час рецензування простежте суму, показану клієнту, суму в експорті та виклик під час повернення коштів. Старі рахунки зберігають суми, записані під час виставлення. Наскрізна перевірка показує, чи до обчислення доходять правильні одиниці. Ізольований тест показує, що обчислення робить із відомими вхідними даними.

У нашій нотатці про продаж останнього квитка рівно один раз такий підхід застосовано до залишків. Нотатка наводить рівність, яка має завжди виконуватися, описує конкурентні тести та пропонує стежити за рівністю в експлуатації. Так рецензент бачить умову, яку має зберегти зміна в будь-якому з відповідних шляхів.

Де виникають проблеми

Вимірювану ознаку легко прийняти за доказ. Покриття показує, які рядки або гілки виконав тест. Тест може пройти, навіть якщо очікуваний результат закріплює хибне правило округлення. Показник складності допомагає вибрати фрагмент для читання. Якщо рознести просте порівняння між кількома незрозумілими функціями, пошук помилки може стати складнішим. Визначте, який дефект має виявляти показник, і розгляньте приклади по обидва боки порога.

Очищення коду теж потребує пріоритетів. Нечітка назва в рідкісному звіті має одні наслідки, неоднозначна умова, яка керує всіма платежами, інші. Зафіксуйте ризик, відокремте потрібні виправлення від можливих поліпшень і визначте, коли переглянути рішення. Суперечку про форматування часто знімає спільне правило й автоматичний форматер.

Локальна зміна може перенести проблему в інше місце. Спільна функція для знижки доречна, якщо кілька викликів справді використовують одне правило. Якщо ж вона вгадує, яка політика діє для продовження, акції чи повернення, код стає менш зрозумілим. З’ясуйте, які дані й умови справді збігаються. Об’єднайте спільну частину, явно покажіть відмінності та перевірте результат.

Що почитати далі

- Правило Google щодо схвалення змін коду
- На що звертають увагу рецензенти Google
- Модульне тестування в Software Engineering at Google

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке якість коду?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/code-quality/