Engineer

Calidad del código

grado en que el código funciona y se entiende y modifica

Qué es

La calidad del código indica en qué medida el código fuente realiza el trabajo previsto y puede entenderse, comprobarse y modificarse con seguridad. Veamos una función de facturación. Una versión arroja el importe correcto para el pedido de demostración, porque ese pedido recorre justo el camino que se tuvo en cuenta al escribirla. Otra identifica la moneda, expresa la regla de redondeo en un solo sitio y distingue el cálculo de un abono del de una venta. Las dos pueden superar la prueba habitual. Para averiguar por qué hay una diferencia en una factura, la segunda ofrece un punto de partida mucho más claro.

La corrección comprende los casos límite, como los datos inválidos, un reintento o un trabajo que queda a medias. La legibilidad se aprecia cuando otra persona entiende las decisiones relevantes sin la ayuda del autor. La facilidad de cambio depende de cuánto se altera al incorporar un requisito y de si se pueden comprobar los efectos. Una función breve puede calcular mal un impuesto. Otra, más enrevesada, puede acertar hoy y hacer muy difícil verificar un cambio de moneda mañana.

Una revisión de código permite evaluar esas propiedades en un cambio concreto. Un control de calidad que condiciona la integración bloquea ciertos fallos detectables. La deuda técnica describe el coste futuro de determinadas decisiones. La calidad del código es el estado del propio código. El criterio de revisión de Google lleva esa distinción a la práctica al pedir que se valore la salud del sistema en conjunto después de cada cambio.

Utilice el concepto en una decisión

Aplique «¿Qué es la calidad del código?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Un equipo que lleva una idea del boceto a un producto al que sus clientes pueden acceder, sin pasarla entre tres proveedores.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Ingresos multiplicados por 10 con mapas GIS — 10x crecimiento de ingresos impulsado desde el producto cartográfico

Ingresos multiplicados por 10 con mapas GIS Póngase en contacto →

Por qué importa

La calidad se percibe sobre todo cuando llega un requisito nuevo. Imaginemos que una aplicación genera presupuestos, emite facturas y tramita devoluciones. Si cada operación calcula por separado el mismo descuento, la siguiente promoción obliga a localizar todas las copias. La discrepancia puede aparecer por primera vez al devolver un pedido. En cambio, si la regla común vive en un lugar conocido y cada operación le pasa sus datos de forma explícita, es más fácil acotar el cambio y decidir dónde probarlo.

La evidencia útil depende del riesgo. Para el dinero, conviene elegir casos en los que redondear cada línea y redondear al final produzcan resultados distintos. En una operación que se puede reintentar, hay que comprobar si una segunda solicitud vuelve a efectuar el cargo. Si un cálculo se ejecuta en todos los pedidos, sería razonable medir el tiempo de respuesta con la carga esperada antes de añadir una caché. Cada observación informa una decisión concreta y tiene un alcance limitado.

También hay un coste de coordinación. Las personas cambian de proyecto y quien revisa una modificación puede no conocer la historia del módulo. Los nombres precisos, un ejemplo ejecutable y una explicación breve de una restricción poco evidente permiten verificar el detalle importante sin depender de la memoria del autor. Esa autonomía ayuda tanto al incorporar a otra persona como al investigar una incidencia fuera del horario habitual.

Cómo se aplica en la práctica

Supongamos que una empresa debe redondear el impuesto sobre el total agrupado por tipo impositivo. En una factura de dos líneas, la cuota de cada una antes del redondeo asciende a 0,005 unidades monetarias. Si se redondea cada cuota al céntimo más próximo, con los empates al alza, el total da 0,02. Si se suman primero y se redondea una vez, da 0,01. El ejemplo permite comprobar la cuenta a mano. La regla de una factura real debe salir de los requisitos del producto y de la normativa aplicable.

El contrato de la función puede indicar en qué unidad trabaja y en qué orden opera. Los importes intermedios necesitan una representación que conserve la precisión requerida. Conviene que el código muestre la agrupación y que el redondeo se aplique en el punto acordado. Una prueba con los importes anteriores distingue las dos reglas. Otra puede mezclar tipos impositivos para evitar que un cambio futuro reemplace los grupos por una suma global. Si los abonos siguen otra regla, hace falta un ejemplo separado.

Al revisar el arreglo, seguimos el importe mostrado a la persona compradora, el que se exporta y el que utiliza la operación de devolución. Las facturas anteriores deben conservar los importes registrados al emitirse. Una prueba de todo el recorrido verifica que quien llama al cálculo le pasa las unidades correctas, algo que una prueba aislada de la función no puede demostrar.

En nuestra nota sobre vender la última entrada una sola vez, el mismo enfoque se aplica al inventario. La nota explicita la relación que deben cumplir las cantidades, propone pruebas con solicitudes concurrentes y propone vigilar esa relación en funcionamiento. Su fórmula permite comprobar qué debe conservarse al modificar cualquiera de las operaciones que afectan a las existencias.

Dónde surgen los problemas

Una señal medible puede confundirse con una garantía. La cobertura muestra qué líneas o ramas se ejecutaron. La prueba seguirá pasando si el valor esperado representa la regla de redondeo equivocada. Una métrica de complejidad ayuda a elegir qué código mirar, pero trocear una comparación clara en varias funciones opacas puede dificultar el diagnóstico. Antes de imponer un umbral, identifiquemos qué defecto real ayudaría a detectar y revisemos casos en ambos lados del umbral.

También se puede invertir tiempo en una limpieza con poco efecto. Un nombre confuso en un informe poco usado tiene un coste distinto del de una condición ambigua que se aplica a todos los cobros. Resulta útil fijar qué riesgo se ha observado, qué correcciones son necesarias y qué mejoras pueden esperar. Una preferencia de formato se puede resolver con una convención compartida y un formateador.

Por último, reducir la duplicación puede ocultar diferencias legítimas. Quizá los descuentos de una renovación y los de una promoción obedecen a políticas distintas, aunque ambos resten dinero. Un método que adivina la política según valores poco relacionados oscurecería ambas operaciones. Vale la pena compartir la parte estable que se entiende y separar las decisiones divergentes. La mejora se aprecia en las pruebas y en la capacidad de otra persona para explicar la regla leyendo el código.

Lecturas adicionales

- Criterio de Google para aprobar cambios de código
- Aspectos que Google comprueba en una revisión
- Pruebas unitarias en Software Engineering at Google

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es la calidad del código?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

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

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por **D.**

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/code-quality/