Engineer

Регресійне тестування

перевірка збереження наявної поведінки після зміни

Що це таке

Регресійне тестування перевіряє, чи зберігає програма після зміни поведінку, яка вже працювала правильно. Нова опція експорту може відповідати своїм вимогам і водночас змінювати рядки, які завантажують наявні користувачі. Перевірка нової опції показує, чи працює ця можливість. Регресійний тест також проходить старим шляхом і перевіряє поведінку, на яку вже хтось спирається.

Посібник NASA описує регресійне тестування як спосіб перевірити, чи зміна не внесла нових дефектів. Серед його порад є повторний запуск тестів, пов’язаних зі зміненими вимогами або кодом, та перевірка модулів, які взаємодіють. Це стосується змін коду, залежностей, налаштувань, даних і середовища. Навіть той самий код може поводитися інакше після оновлення браузера.

Перевірка виправлення і перевірка регресій доповнюють одна одну. Спершу відтворіть помилку й перевірте виправлений результат. Потім перевірте пов’язану поведінку, яка має зберегтися. Випадок початкової помилки можна залишити в тестах як попередження про повернення дефекту. Очікування має відбивати чинну домовленість. Якщо вимогу змінено свідомо, оновіть результат і поясніть, що зміниться для попередніх викликів.

Застосуйте поняття до рішення

Застосуйте «Що таке регресійне тестування?» 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

Наслідки часто виходять за межі виправлених рядків. Заміна списку множиною може прибрати дублікати і порушити порядок звіту. Розбір дати може прийняти новий формат і відхилити порожнє значення, яке надходить з іншої частини продукту. Автоматизований тест пов’язує конкретний наслідок із недавньою зміною. Це особливо важливо для інтеграції, яку розробник рідко запускає локально.

Для невеликої зміни почніть із найближчого спільного правила та кількох різних реальних викликів. Запитайте, чи зміняться типові налаштування, граничні випадки, оброблення помилок або вже збережені дані. Для важливих змін можуть знадобитися і перевірки швидкодії та доступності. Рівень доказу має відповідати наслідкам пропущеної помилки. Модульного тесту може вистачити для перетворення значення. Перевірка роботи з клавіатури може потребувати справжнього браузера та фактичного шляху користувача.

Назва тесту пояснює, навіщо він існує. «Пропущена дата лишається порожньою під час експорту старих справ» говорить більше, ніж «третій приклад пройшов». Покажіть істотні дані й потрібний результат. Наступний розробник зможе відрізнити навмисну зміну очікування від поломки наявної можливості.

Як це виглядає на практиці

Уявімо експорт CSV з опцією включати архівні облікові записи. Без опції вони мають лишитися поза файлом. Активні записи залишаються відсортованими за ідентифікатором, а відсутня дата останнього контакту лишається порожньою. Опція додає архівні записи без дублювання активних рядків. Підготуйте два активні записи у зворотному порядку й один архівний. Для одного активного пропустіть дату. Запустіть обидва шляхи через справжній експорт.

Для типового режиму порівняйте ідентифікатори, порядок і порожню клітинку. Коли опцію ввімкнено, ті самі активні та архівний запис мають з’явитися по одному разу. Додайте ім’я з комою. Експорт, який лише склеює текст, може створити потрібну кількість рядків, але рознести значення по неправильних стовпцях. Прочитайте файл інструментом такого самого типу, яким користується одержувач.

Перевірте, чи тести відхилять справжню регресію. У тестовій версії тимчасово включіть архівні записи до типового експорту або приберіть лапки навколо імені з комою. Для кожної помилки має спрацювати зрозуміла перевірка. Відновіть справну версію і запустіть тести ще раз. Перевірка самої наявності файла може пропустити обидві помилки. Додатково перевірте коректний випадок без архівних записів.

Наша нотатка про підкреслення і підмножини шрифтів у Safari показує іншу межу доказу. Зміна CSS начебто виправила відображення у Chromium, але Safari досі малював лінію через нижні виносні елементи літер. Регресійний тест такого явища має спостерігати роботу потрібних браузерів у потрібних умовах. У нотатці описано діагноз і спостереження. Тест допоміг би зберегти цей висновок після наступних змін.

Де виникають проблеми

Кодування файла теж може бути частиною домовленості. Якщо одержувач читає українські імена, додайте показовий запис і відкрийте результат тим самим типом програми. Однакова кількість рядків не помітить пошкоджених літер. У разі зміни формату зробіть окремий приклад для попереднього способу читання, поки його ще підтримують.

Велика кількість тестів може створювати оманливе враження повноти. Сотні малих перевірок можуть не охопити єдину інтеграцію, що читає старі дані. Добирайте приклади за реальними діями, наявними даними і зачепленими взаємодіями. Якщо помилка вже пройшла непоміченою, захистіть постраждалий шлях і перевірте, що тест відхиляє стару помилку.

Можна також порівняти зайві деталі. Повний знімок зміниться через випадкові ідентифікатори або годинник. Порівнюйте порядок, коли його обіцяно. Якщо гарантовано лише набір, зіставте його склад і окремо перевірте відсутність дублікатів. Переглядайте обґрунтування навмисних змін очікувань так само уважно, як код.

Нестабільна перевірка ускладнює тлумачення збоїв. Чекайте спостережуваної умови з обмеженням часу. Використовуйте відомі дані й прибирайте стан між спробами. Зберігайте широкі перевірки для місць, де справді важлива взаємодія, і швидкі перевірки поблизу розробки. Вони дають докази на різних рівнях.

Що почитати далі

- Посібник NASA з регресійного тестування програмного забезпечення
- Документація pytest про параметризацію тестів
- 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 регресійне тестування?» 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/regression-testing/