Engineer

Regressionstest

Prüfung, ob eine Änderung bisheriges Verhalten erhält

Was es ist

Ein Regressionstest prüft, ob Software nach einer Änderung weiterhin das vorherige korrekte Verhalten zeigt. Eine neue Exportoption kann genau nach Anforderung funktionieren und trotzdem die Zeilen ändern, die bisherige Nutzer erhalten. Ein Test der neuen Option sagt, ob die gewünschte Funktion arbeitet. Ein Regressionstest durchläuft auch den bisherigen Weg und prüft, ob das Verhalten erhalten bleibt, auf das sich andere verlassen.

Das Handbuch der NASA erläutert Regressionstests als Nachweis dafür, dass Änderungen keine neuen Fehler eingeführt haben. Dazu kann man vorhandene Tests wiederholen, die auf geänderten Code oder Anforderungen zurückgehen, und beteiligte Module berücksichtigen. Der Gedanke gilt bei Änderungen am Quellcode, an Abhängigkeiten, Einstellungen, Daten oder der Umgebung. Auch unveränderter Code kann sich nach einem Browserwechsel anders verhalten.

Die Bestätigung einer Korrektur und Regressionstests ergänzen sich. Zuerst wird der Fehler reproduziert und das korrigierte Verhalten bestätigt. Anschließend wird geprüft, ob benachbarte Funktionen weiterhin richtig arbeiten. Das ursprüngliche Fehlerbeispiel kann im Testbestand bleiben, damit eine Wiederkehr auffällt. Der Sollwert muss die versprochene Funktion ausdrücken. Ändert sich eine Anforderung absichtlich, wird die Erwartung angepasst und festgehalten, welche Aufrufer davon betroffen sind.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Regressionstest?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Ein Team führt eine Idee von der Skizze bis zu einem Produkt, bei dem Ihre Kunden sich anmelden können, statt sie zwischen drei Dienstleistern weiterzureichen.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. GIS-Kartenprodukt zu 10-fachem Umsatz geführt — 10x Umsatzwachstum durch die Leitung des Kartenprodukts

GIS-Kartenprodukt zu 10-fachem Umsatz geführt Kontakt aufnehmen →

Warum es wichtig ist

Die Wirkung einer Änderung reicht oft über die bearbeiteten Zeilen hinaus. Eine Menge statt einer Liste kann Duplikate entfernen und die Reihenfolge eines Berichts ändern. Eine Datumsfunktion kann ein neues Format akzeptieren und an anderer Stelle erlaubte leere Werte abweisen. Ein automatisierter Test hilft, eine konkrete Folge mit einem frischen Eingriff in Verbindung zu bringen. Das zählt besonders für Integrationen, die bei der Entwicklung kaum jemand lokal startet.

Beginnen Sie bei kleinen Änderungen mit der nächsten gemeinsamen Regel und mit wenigen wirklichen Aufrufern, die sich unterscheiden. Fragen Sie, ob Standardwerte, Grenzen, Fehlerbehandlung oder schon gespeicherte Daten betroffen sind. Bei Änderungen mit großen Folgen können Leistung und Barrierefreiheit ebenso wichtig sein wie einzelne Werte. Der Nachweis muss zur Auswirkung eines übersehenen Fehlers passen. Eine Umrechnung lässt sich mit einem Modultest genau prüfen. Die Tastaturbedienung kann einen Browserdurchlauf entlang des tatsächlichen Benutzerwegs erfordern.

Ein benanntes Beispiel erklärt, warum eine Regel erhalten bleiben soll. „Fehlendes Datum bleibt beim Export älterer Vorgänge leer“ sagt mehr als „Fall drei besteht“. Zeigen Sie die wichtigen Eingabewerte und das Ergebnis. Die nächste Person kann dann sehen, ob sich die Erwartung absichtlich ändert oder eine bisher benutzte Funktion kaputtgegangen ist.

Wie es in der Praxis aussieht

Angenommen, ein CSV-Export erhält die Option, archivierte Konten einzubinden. Standardmäßig bleiben diese ausgeschlossen. Aktive Konten stehen weiterhin nach Kennung sortiert. Fehlt das Datum des letzten Kontakts, bleibt die Zelle leer. Die neue Option fügt Archive hinzu, ohne aktive Datensätze zu verdoppeln. Verwenden Sie zwei aktive Konten in umgekehrter Reihenfolge und ein archiviertes Konto. Lassen Sie bei einem Konto das Datum weg. Führen Sie beide Varianten über den tatsächlichen Export aus.

Vergleichen Sie im Standardfall die Kennungen, ihre Reihenfolge und das leere Datumsfeld. Mit gesetzter Option erscheinen die aktiven Konten und das archivierte je einmal. Fügen Sie einen Namen mit Komma hinzu. Ein Export, der einzelne Texte nur verbindet, kann die richtige Zeilenzahl mit falschen Spalten liefern. Lesen Sie die CSV-Datei mit einem vergleichbaren Programm wie der Aufrufer. Das Beispiel soll überschaubar sein und zugleich die Wirkung sichtbar machen.

Prüfen Sie, ob die Tests eine echte Regression erkennen. Lassen Sie versuchsweise archivierte Konten auch ohne Option ausgeben oder entfernen Sie die CSV-Quotierung beim Namen mit Komma. Die betroffenen Prüfungen sollten mit verständlichen, unterschiedlichen Gründen fehlschlagen. Stellen Sie die korrekte Fassung wieder her und wiederholen Sie die Prüfung. Eine Prüfung, die nur das Vorhandensein einer Datei verlangt, würde beide Fehler leicht übersehen. Ein Fall ohne archivierte Konten schützt außerdem die gültige leere Auswahl.

Unsere Notiz über Unterstreichungen und Schriftteilsätze in Safari zeigt eine andere Grenze. Eine CSS-Änderung schien die Darstellung in Chromium zu korrigieren. Safari zeichnete die Linie weiter durch Unterlängen. Ein Schutz gegen diesen Darstellungsfehler müsste die behaupteten Browser und Situationen tatsächlich beobachten. Die Notiz beschreibt die Diagnose und die Versuche. Der vorgeschlagene Test würde die Erkenntnis absichern.

Wo es schiefgeht

Viele Prüfungen können eine falsche Sicherheit vermitteln. Hunderte Einzelvergleiche übersehen womöglich den einzigen Aufrufer, der Altdaten liest. Wählen Sie Fälle anhand tatsächlicher Abläufe, vorhandener Eingaben und betroffener Wechselwirkungen. Ist ein Fehler schon einmal durchgerutscht, schützen Sie den betroffenen Ablauf und prüfen Sie, ob das unerwünschte Verhalten einen Fehler auslöst.

Auch ein zu umfassender Vergleich führt in die Irre. Eine vollständige Momentaufnahme ändert sich mit zufälligen Kennungen oder Uhrzeiten. Ist die Reihenfolge versprochen, prüfen Sie sie. Sonst genügt ein Vergleich der Menge mit einer zusätzlichen Kontrolle auf Duplikate. Begründen und begutachten Sie Änderungen am erwarteten Ergebnis so sorgfältig wie eine Änderung am Code.

Instabile Tests erschweren schließlich die Einordnung von Fehlern. Warten Sie mit Frist auf einen beobachtbaren Zustand, vermeiden Sie Wartezeiten nach Gefühl und bereiten Sie bekannte Daten vor. Ein Test darf keine unerwarteten Zustände für den nächsten hinterlassen. Nahe am Code bleiben schnelle Prüfungen sinnvoll. Für echte Wechselwirkungen bleiben umfassende Prüfungen wichtig.

Weiterführende Lektüre

- NASA-Handbuch zum Regressionstest von Software
- pytest-Dokumentation zur Parametrisierung von Prüffällen
- Unit-Tests in Software Engineering at Google

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Regressionstest?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/regression-testing/