Engineer

Regressionstestning

testning av att en ändring bevarar tidigare beteende

Vad det är

Regressionstestning undersöker om programvara bevarar beteende som tidigare fungerade efter en ändring. En ny exportinställning kan uppfylla alla nya krav men samtidigt ändra vilka rader befintliga användare får. Ett test av den nya inställningen visar om den fungerar. Ett regressionstest går också igenom det tidigare flödet och kontrollerar om beteendet som andra litar på finns kvar.

NASA:s handbok beskriver regressionstestning som ett sätt att kontrollera att en ändring inte har infört nya fel. Råden omfattar att köra befintliga tester som hör ihop med ändrad kod eller ändrade krav, och att ta hänsyn till moduler som påverkar varandra. Samma idé gäller vid ändringar i beroenden, inställningar, data och körmiljö. En uppdaterad webbläsare kan ändra beteendet även när koden står stilla.

Att bekräfta en rättelse och att undersöka regressioner är två uppgifter som passar ihop. Återskapa felet och verifiera det rättade resultatet. Prova sedan det närliggande beteende som behöver bevaras. Det ursprungliga fallet kan ingå i testuppsättningen och varna om felet återkommer. Förväntningen ska beskriva vad produkten lovar. Ändras kravet avsiktligt, uppdatera förväntningen och förklara effekten för de anropare som redan finns.

Använd begreppet i ett beslut

Använd ”Vad är regressionstestning?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Ett team som tar en idé från en skiss till något dina kunder kan logga in i, i stället för att lämna den vidare mellan tre leverantörer.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Ledde GIS-kartappen till 10 gånger högre intäkter — 10x intäktstillväxt, med kartprodukten i spetsen

Ledde GIS-kartappen till 10 gånger högre intäkter Hör av er →

Varför det spelar roll

Följden av en ändring når ofta längre än de redigerade raderna. Om en mängd ersätter en lista kan dubbletter försvinna och en rapport få annan ordning. En datumfunktion kan acceptera ett nytt format och sedan avvisa tomma värden från en annan del av produkten. Ett automatiserat test kopplar en viss följd till en färsk ändring. Det är särskilt viktigt för den integration som utvecklaren sällan kör på sin dator.

Börja vid en liten ändring med närmaste gemensamma regel och några olika, verkliga anropare. Fråga hur standardvärden, gränser, felhantering eller redan sparade data påverkas. Vid allvarligare förändringar kan tillgänglighet och prestanda vara lika viktiga som enskilda värden. Underlaget behöver passa konsekvensen av att missa felet. Ett enhetstest kan kontrollera en omvandling exakt. Tangentbordsbeteende kan kräva en riktig webbläsare längs samma väg som besökaren tar.

Ett tydligt namn hjälper den som förvaltar testet att förstå varför det finns. ”Saknat datum förblir tomt vid export av äldre fall” ger mer än ”testfall tre godkänt”. Visa de väsentliga värdena och resultatet. Den som läser testet kan bedöma om en förändrad förväntning följer av ett nytt krav eller av en oavsiktlig försämring.

Så ser det ut i praktiken

Tänk dig en CSV-export med ett val för arkiverade konton. Utan valet är sådana konton fortsatt uteslutna. Aktiva konton visas i ordning efter referens. En saknad uppgift om senaste kontakt förblir tom. Valet lägger till arkiverade konton utan att dubblera aktiva rader. Lägg in två aktiva konton i omvänd ordning och ett arkiverat konto. Lämna datumet tomt för det ena kontot. Kör båda vägarna genom den verkliga exporten.

Jämför referenser, ordning och den tomma cellen utan valet. Med valet på ska samtliga konton finnas en gång var. Lägg till ett namn som innehåller ett kommatecken. En export som bara sammanfogar text kan ge rätt antal rader

med fel innehåll i kolumnerna. Läs CSV-filen med samma sorts läsare som en riktig mottagare använder. Exemplet ska vara litet men kunna visa effekten.

Prova om testerna avvisar ett verkligt fel. Låt arkiverade konton ingå som standard en stund eller ta bort citattecknen runt fältet med kommatecken. Kontrollerna ska ge olika och begripliga fel. Återställ den fungerande ändringen och kör igen. En kontroll som enbart letar efter filen kunde ha missat båda felen. Prova också ett giltigt fall där det inte finns några arkiverade konton.

Vår anteckning om understrykningar och delmängder av teckensnitt i Safari visar en annan gräns för underlaget. En ändring av CSS verkade lösa problemet i Chromium samtidigt som Safari fortfarande drog strecket genom bokstävernas nedstaplar. Ett regressionstest skulle behöva observera de faktiska webbläsarna och situationerna. Anteckningen dokumenterar diagnosen och observationerna. Testet vore ett sätt att bevara insikten.

Var det går fel

Filens egenskaper är ibland en del av avtalet. En mottagare som läser ett visst teckenkodningsformat behöver kunna tolka namn med bokstäver utanför grundalfabetet. Ta bara med den kontrollen när formatet är viktigt för mottagaren, och verifiera resultatet med en faktisk läsare. Ett radantal fångar inte ett trasigt namn.

Många testfall kan ge en överdriven bild av täckningen. Hundratals små kontroller kan missa den enda integrationen som läser historiska poster. Välj fall utifrån faktiska användare, viktiga indata och berörda samspel. Om ett fel har gått obemärkt förbi, skydda det drabbade flödet och kontrollera att testet avvisar det felaktiga beteendet.

Det går också att jämföra för mycket. En komplett ögonblicksbild ändras med klockan eller slumpade identiteter. Kontrollera ordningen om den utlovas. Om bara urvalet utlovas, jämför mängden och leta efter dubbletter separat. Granska avsiktliga ändringar av det förväntade resultatet noggrant.

Ett instabilt test gör slutligen fel svårtolkade. Vänta på ett observerbart tillstånd med en tidsgräns, använd kända data och städa efter varje försök. Låt snabba kontroller följa utvecklingen och behåll bredare provningar där verkliga samspel spelar roll. Kombinationen bidrar med belägg på flera nivåer.

Läs mer

- NASA:s handbok om regressionstestning av programvara
- pytest-dokumentationen om parametrisering av testfall
- Enhetstestning i Software Engineering at Google

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är regressionstestning?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/regression-testing/