Engineer

Regressionstest

test af om en ændring bevarer eksisterende adfærd

Hvad det er

En regressionstest kontrollerer, at software efter en ændring bevarer adfærd, der allerede virkede. En ny eksportindstilling kan opfylde alle sine krav og stadig ændre de rækker, en eksisterende bruger får. Prøver man alene den nye indstilling, får man svar på, om den nye funktion virker. En regressionstest gennemgår også det tidligere forløb og undersøger, om en vigtig egenskab holder.

NASA's håndbog beskriver regressionstest som en metode til at kontrollere, om ændringer har medført nye fejl. Den peger på at gentage kendte kontroller, der er knyttet til ændret kode eller ændrede krav, og på at inddrage de moduler, der har forbindelse til hinanden. Ideen gælder ved opdatering af kode, afhængigheder, opsætning, data eller miljø. Et program kan ændre adfærd, selv om kildekoden står urørt.

Bekræftelse af en rettelse og regressionstest supplerer hinanden. Genskab først fejlen, og kontrollér det rettede resultat. Undersøg så den nærliggende adfærd, der skal bevares. Det oprindelige fejltilfælde kan blive i testsættet, så en gentagelse bliver synlig. Det forventede resultat skal beskrive, hvad produktet lover. Når en ændring er bevidst, så opdatér forventningen og angiv de eksisterende anvendelser, der nu skal følge et andet forløb.

Brug begrebet i en beslutning

Brug »Hvad er en regressionstest?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Ét team, der tager en idé fra skitse til noget, jeres kunder kan logge ind på, i stedet for at kaste den mellem tre leverandører.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har ført en GIS-kortapp til 10x omsætningsvækst — 10x omsætningsvækst, drevet af kortproduktet

Har ført en GIS-kortapp til 10x omsætningsvækst Kontakt os →

Hvorfor det betyder noget

Virkningen når ofte længere end den redigerede kode. En mængde i stedet for en liste kan fjerne dubletter og flytte rækkefølgen i en rapport. En datofunktion kan forstå et nyt format og afvise et tomt felt accepteret i en anden del af programmet. En automatiseret test gør det muligt at knytte en konkret følge til den aktuelle ændring. Det er særligt nyttigt for integrationer, som udvikleren sjældent kører lokalt.

Ved en lille ændring kan man begynde med den nærmeste fælles regel og et par virkelige, forskellige kald. Undersøg, om standardvalg, grænser, fejltilstande eller allerede gemte data kan blive påvirket. Ved større ændringer kan svartid eller tilgængelighed være lige så vigtigt som de enkelte værdier. Vælg et bevis, der passer til konsekvensen af en overset fejl. En enhedstest kan fastslå en omregning præcist. For tastaturbetjening kan det kræve en rigtig browser og den vej, en bruger tager.

Eksemplets navn kan forklare, hvorfor reglen bevares. »Tom dato forbliver tom, når gamle sager eksporteres« fortæller mere end »forløb tre består«. Vis de værdier, der har betydning, og det forventede resultat. Den næste udvikler kan da bedømme, om et nyt krav tilsigtet ændrede reglen, eller en fejl har ramt det gamle forløb.

Sådan ser det ud i praksis

Forestil dig en CSV-eksport med et nyt valg: medtag arkiverede konti. Standardeksporten skal fortsat udelade dem. Aktive konti holder rækkefølgen efter kontoreference, og et manglende felt for seneste kontakt er stadig tomt. Det nye valg tilføjer arkiver uden at gentage aktive rækker. Sæt to aktive konti i omvendt rækkefølge og én arkiveret konto ind i prøvedataene. Lad datoen mangle for én af de aktive. Kør begge forløb gennem den virkelige eksport.

Sammenlign referencerne, deres rækkefølge og det tomme datofelt i standardforløbet. Med valget slået til skal hver konto forekomme én gang. Medtag et navn med et komma. En eksport, der blot lægger tekst sammen, kan

ramme det rigtige antal rækker og stadig placere værdier i forkerte kolonner. Læs filen med samme type læser som en reel modtager. Hold eksemplet lille nok til at gennemgå.

Kontrollér, at testen opdager en faktisk regression. Lad midlertidigt standardeksporten medtage arkiver, eller fjern anførselstegnene i feltet med komma. Kontrollerne skal give forskellige og tydelige fejl. Gendan den korrekte kode, og kør igen. En kontrol af, om der findes en fil, kunne overse begge fejl. Et forsøg uden arkiverede konti beskytter desuden den gyldige tomme udvælgelse.

Vores note om understregninger og skriftdelinger i Safari viser, at et forsøg har en grænse. En ændring af CSS så ud til at rette Chromium, mens Safari stadig trak linjen gennem bogstavernes nedstreger. En test af den adfærd skal se den virkelige gengivelse i de browsere og situationer, den dækker. Noten viser diagnosen og de faktiske observationer. En regressionstest kan fastholde det, som undersøgelsen lærte os.

Hvor det går galt

Antallet af tests siger ikke, hvilke forløb de dækker. Mange enkle kontroller kan stå grønne, selv om ingen kører den integration, der læser gamle data. Vælg forsøg efter faktiske arbejdsgange, inddata og berørte afhængigheder. Hvis et problem slap igennem, så beskyt det konkrete forløb og efterprøv, at testen afviser den gamle fejl.

Man kan også sammenligne for meget. Et komplet øjebliksbillede ændrer sig med tilfældige identifikatorer eller tidspunkter. Hvis rækkefølgen er en aftale, bør testen bevare den. Hvis aftalen kun handler om et sæt poster, så sammenlign sættet og se særskilt efter dubletter. Gennemgå en tilsigtet ændring af forventningen lige så omhyggeligt som en kodeændring.

Til sidst gør ustabile tests fejl svære at fortolke. Vent på en observerbar tilstand med en tidsgrænse, brug kendte data, og ryd tilstanden op mellem forsøg. Behold større kontroller dér, hvor virkelige samspil skal undersøges. Små, hurtige kontroller kan ligge tættere på udviklingen. Kombinationen giver beviser på flere niveauer.

Læs mere

- NASA's håndbog om regressionstest af software
- pytest-dokumentationen om parametrisering af testtilfælde
- Enhedstest i Software Engineering at Google

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en regressionstest?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/regression-testing/