Engineer Programvare

Testing og kvalitetssikring

29 resultater

Knytt sammen delene av systemet

Arbeidet under «Testing og kvalitetssikring» viser én del av et fungerende system. Vi kan hjelpe med å knytte den delen til resten av driften deres. Delen brukerne aldri ser: hvor dataene ligger, hvilke regler som gjelder, og hvordan andre systemer får lov til å snakke med den.

Se tjenestene våre Ta kontakt →

Finn en sammenlignbar utfordring

Åpne et eksempel som ligner situasjonen deres. Gjennomgangen forklarer beslutningene bak resultatet, utover navnet på teknologien. Designet og driftet 6 ETL/ELT-pipelines — 6 ETL-pipelines på tvers av tre databasemotorer

Designet og driftet 6 ETL/ELT-pipelines Ta kontakt →

Migrerte API-et til Huma v2 med 100 % paritet

Migrerte HTTP-API-et fra Fiber til Huma v2 — 649 stier og 760 operasjoner — og nådde og holdt 100 % samsvar mellom rutene serveren registrerer og OpenAPI-beskrivelsen den publiserer.

Genererte API-kontrakten med databasen som kilde

Genererte API-kontrakten fra databasen og utover — OpenAPI, en typet TypeScript-klient på 44 076 linjer, 61 mock-handlere og grensene brukergrensesnittet håndhever — med en kontroll i hvert ledd som feiler ved avvik.

Bygde en testsuite for Go, SQL og nettleser

Bygde en lagdelt automatisert testsuite — 981 Go-tester, 543 spesifikasjoner for frontend og nettleser, 494 atferdstester i SQL — med mutasjonstesting, egenskapsbaserte tester og en kvalitetsport for universell utforming.

Bygde repoets kontrollmotor med 268 kontroller

Bygde repoets kontrollmotor — 268 registrerte commit-kontroller, 277 lint-regler og 15 egne ESLint-regler — pluss 146 tester av selve kontrollene, slik at det er bygget, ikke gjennomgangen, som holder standarden.

Bygde en commit-port med 22 lintere

Bygde en commit-port med 22 enlinjes lintere pluss fem som fortjener et avsnitt, uten advarselsnivå og uten tillatte undertrykkinger i koden, som dekker HTML, CSS, JavaScript, Python, YAML, Markdown, shell, lenker, staving, hemmeligheter og typografi.

Et sikkerhetsnett som ikke lenger koster en halvtime

Kuttet nettstedets nettleserdrevne kvalitetsport fra 1 636 sekunder til 615 ved å kjøre kontrollene med de lengste først gjennom en worker-pool begrenset til fire baner, etter å ha målt at alfabetisk rekkefølge kostet 320 sekunder mot 224.

Byttet 27 skriftstørrelser mot en skala i 6 trinn

Erstattet 27 gjengitte skriftstørrelser, der de nærmeste naboene lå 0,6 % fra hverandre, med en Major Third-skala i seks trinn, og skrev linteren som avviser den tjueåttende.

Tilgjengelighetskontroller på 32 sider i begge temaer

Implementerte WCAG 2.2 Level AA-kontroller på 32 representative sider i begge fargetemaer, med en samsvarsoversikt, kontroll av ett AAA-kriterium og det delvise resultatet for kriterium 1.4.13 uttrykkelig angitt.

Feil i universell utforming som kontrollverktøyet deres overser

Fant og rettet 11 feil i universell utforming som nettleseren meldte var i orden — åtte bunntekstlenker som fortsatt lå i tabulatorrekkefølgen bak pointer-events, en scroll-tidslinje som klippet kolofonen av tre sider, og en regelmotor som kjørte 70 av sine 105 regler.

Lukket fargesystemet ved 28 farger

Lukket fargesystemet ved 28 dokumenterte farger med en linter som avviser en verdi som er malt, men udokumentert, dokumentert, men ikke malt, feilmålt, skrevet om som en literal, eller innenfor en perseptuell avstand på 0,02 fra en som allerede finnes.

Fant at det lyse temaet manglet et helt lag

Fant ut at det lyse temaet hadde manglet et helsides overleggslag helt siden det ble skrevet, ved å kreve at begge temaene maler hver lagdelte flate med like mange lag.

Gjorde innstillingen for redusert bevegelse ærlig

Gjorde innstillingen for redusert bevegelse ærlig etter å ha funnet elleve selektorer som fortsatt animerte under den, fordi en universell transition-none-regel taper på spesifisitet mot enhver regel med en klasse.

Rettet et sitemap med ett felles tidsstempel

Rettet et sitemap der 172 av 176 URL-er delte ett endringstidspunkt, ved å hente datoen fra git-historikken etter å ha slått fast at eksporten skriver om hver fil ved hver kjøring.

Lintet selve kvalitetsporten på 10 242 linjer

Satte 10 242 linjer JavaScript i kvalitetsporten under en formaterer og en linter etter å ha slått fast at det var den største kodemengden i repoet og den eneste ingenting leste, og rettet 13 funn uten å undertrykke noen.

Holdt generatoren til 981 tester og 92 % dekning

Holdt generatoren til 981 testtilfeller med et minstekrav på 92 % grendekning og advarsler behandlet som feil, og verifiserte idempotens ved å kjøre hele byggepipelinen to ganger fra en tom fil og kreve at den andre kjøringen ikke endret noe.

Bestod 106 av 106 PDF/UA-1-profilkontroller

Dokumenterte PDF/UA-1-validering av ni dokumenter med 106 av 106 bestått kontroller i den konfigurerte profilen for universell utforming, mens arkivvarianten feilet på én av 146 regler.

Aktiverte alle linterregler som feil

Slo på alle reglene i Python-linteren på feilnivå og arbeidet seg gjennom 1 815 funn ned til null, med en skriftlig begrunnelse for hvert av de få unntakene og to av dem sikret av en kontroll i stedet for en kommentar.

Bygde inn en QR-enkoder på 522 linjer

Bygde inn en QR-enkoder — Reed-Solomon over GF(256), fast moduloppsett, åtte maskemønstre — på 522 linjer i stedet for å ta inn en tredje kjøretidsavhengighet, og verifiserte den ved å lese den ferdige matrisen tilbake med en uavhengig skrevet dekoder.

Skrev tester for selve kontrollverktøyene

Skrev tester for selve kontrollene etter å ha slått fast at en kontroll som bare får ren inndata, en dag vil melde rent fordi den ikke leste noe — plantet en stavefeil for å bekrefte at stavekontrollen finner den, og hentet et id-område fra databasen i stedet for fra et tall i testen.

Bygde en tallkontroll på tvers av språk

Bygde en innholdskontroll på tvers av språk som feiler når en oversettelse mister et tall den engelske teksten oppgir, og når et språk bruker en notasjon som ikke hører hjemme i det — og fant to danske beskrivelser som manglet en måleverdi, og seksten ukrainske tekstbiter med engelske anførselstegn.

En avgrenset hendelsesflyt i en macOS-app

Avgrenset hendelsesbuffere og logging etter en registrert macOS-hendelse med vekst på 41 MB i sekundet og 111 GB komprimerte sider på en maskin med 36 GB; en registrert full testkjøring reduserte kjernens logglinjer fra 610 996 til 1 411.

Tok i bruk streng samtidighet i Swift 6 over C

Tok i bruk fullstendig streng samtidighet i Swift 6 uten actors, og koblet en blokkerende C-hendelsesløkke til main actor gjennom én produsent, én konsument og én rekkefølge — etter å ha slått fast at én task per hendelse mister rekkefølgen grensesnittet er avhengig av.

Skrev en parser som verifiserer hvert FFI-kall

Skrev en parser som leser den faktiske C-headeren på 7 308 linjer og verifiserer hvert kallsted, hver enum-konstant og at hver klasse som eier en peker, er final, etter at en håndskrevet plassholder-header lot kall til tre fjernede funksjoner kompilere, lenke og krasje.

Ga alle 17 ikoner navn for skjermlesere

Ga hver kontroll med bare ikon i grensesnittet et navn for skjermlesere etter å ha funnet at send-knappen ble lest opp som «arrow up circle, button», og skrev linteren som krever etiketten innen åtte linjer fra ikonet.

Bygde et macOS-designsystem med 48 tokens

Bygde et designsystem med 48 tokens på plattformens eget glassmateriale og skrev linteren som avviser et magisk tall, en hardkodet skriftstørrelse eller en animasjon som ignorerer innstillingen for redusert bevegelse.

Nådde den ubrukte halvdelen av kjernen

Tok i bruk den halvdelen av meldingskjernen applikasjonen aldri hadde brukt — overføring av sikkerhetskopier, forsvinnende meldinger, redigering og ny sending av meldinger, verifiserte invitasjoner, proxyer og krypteringspolicy — og kjørte hver test mot det ekte biblioteket uten mocks.

Skrev om feilmeldingene som skyldte på brukerne

Skrev om applikasjonens feilmeldinger i tråd med en skriftlig tonestandard etter at en avvist innlogging ga brukeren skylden for en skrivefeil når leverandøren egentlig krevde et appspesifikt passord, og dekket det med en test som navngir leverandøren.

Bygde den visuelle identiteten fra 2 tegninger

Bygde selskapets ikon- og favicon-sett fra to håndlagde tegninger, med alle avledede filer regenerert av skript og en kontroll som feiler når en avledet fil ble committet før kilden.

Fikk seks innsamlere til å si hva de hoppet over

Fikk seks rapporteringsinnsamlere til å oppgi den delen av inndataene de aldri leste — filer som forble lukket, sonder som ikke var installert, journallinjer som ikke ble gjenkjent — og tilpasset terskelverdier til 116 målinger, etter å ha slått fast at en blind innsamling og en vert der alt er i orden, gir den samme siden.

Fokuser den tekniske innsatsen

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Sett emnet i deres sammenheng

Fortell hvordan «Testing og kvalitetssikring» inngår i systemet deres. Vi kan hjelpe med å velge riktig tjeneste og avgrense arbeidet.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://software.engineer.company/nb/categories/testing/