Engineer

Defekt

en mangel, der gør, at et produkt ikke opfylder kravene

Hvad det er

En defekt er noget forkert i selve et stykke arbejde, som får arbejdet til at opføre sig anderledes end tiltænkt eller til at udelade noget, det skulle have gjort. Til hverdag siger de fleste udviklere »bug« eller bare »fejl«, og i testlitteraturen møder man også de engelske »fault« og »flaw«.

Det afgørende ord er *arbejde*. En defekt kan sidde i kildekoden, men lige så vel i et krav, der modsiger et andet krav, et design, der ikke kan bære den belastning, det blev tegnet til, en konfigurationsfil, en test, der forventer den forkerte værdi, eller en vejledning, der beskriver en knap, produktet fjernede for to versioner siden. Uanset hvor den sidder, hører defekten til selve tingen, og den er der, hvad enten nogen har bemærket den eller ej.

På dansk dækker »fejl« alle tre led i en kæde, som testfaget gerne holder adskilt. Et menneske begår en fejltagelse: læser en specifikation forkert, går ud fra noget forkert eller rammer den forkerte tast. Fejltagelsen efterlader en defekt i det, personen lavede. Når den defekte del kører under forhold, der afslører den, opfører produktet sig forkert, og den synlige forkerte opførsel er et svigt (failure). Kæden går kun den ene vej og kan standse ved ethvert led. En defekt i en gren, der aldrig bliver udført, eller en, der kræver et input, ingen sender, kan ligge i en kodebase i årevis uden at give et eneste svigt.

Hvad der tæller som forkert, afhænger af, hvad man forventede, og forventninger rækker længere end den skrevne specifikation. Produktkvalitetsmodellen i ISO/IEC 25010 beskriver funktionel egnethed (functional suitability) som den grad, hvori et produkt leverer funktioner, der opfylder udtrykte og underforståede behov, når det bruges under bestemte betingelser. Egnetheden deles i fuldstændighed (dækker funktionerne de opgaver og mål, brugerne har), korrekthed (er resultaterne nøjagtige) og

hensigtsmæssighed (hjælper funktionerne med at få opgaven løst). En forkert sum er et problem med korrektheden; en rapport, man ikke kan eksportere, et problem med fuldstændigheden. Pålidelighed står i modellen som en egenskab for sig med blandt andet fejlfrihed, tilgængelighed (availability), fejltolerance og genoprettelighed som underpunkter, så et produkt, der regner rigtigt og derefter går ned to gange om dagen, stadig har defekter.

En regression er en defekt med en forhistorie: noget, der virkede, og som en senere ændring ødelagde. Den har fået sit eget navn, fordi den kommer med et spor, som en oprindelig defekt mangler, nemlig en version, man ved var god.

Brug begrebet i en beslutning

Brug »Hvad er en defekt?« 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

Hvad en defekt koster, afhænger mest af, hvor den bliver opdaget. Fanger den, der skriver koden, den selv, koster det et minut og efterlader intet spor. Fanges den i review, koster den en samtale. Fanges den af en kunde, koster den selve svigtet, supporthenvendelsen, fejlsøgningen, rettelsen og noget tillid, og har den gjort reel skade, er den blevet en del af en driftshændelse. Typesystemer, lintere, test og kodereview arbejder alle på den samme strækning og prøver at rykke opdagelsen tættere på det øjeblik, defekten blev skrevet.

Defekter kommer også i familier. Fejltagelsen bag én har som regel slægtninge: den samme misforståede grænseflade kaldt på samme måde seks steder, den samme forveksling af enheder kopieret ind i hvert nyt modul, den samme ukontrollerede antagelse i hver request-handler. Det frugtbare spørgsmål efter en defekt er mindre, hvor den forkerte linje står, og mere, hvad der gjorde linjen let at skrive, og hvor det samme ellers er sket.

Og defekter vender tilbage. Kode ændres hele tiden, og en rettet defekt kan komme igen gennem en refaktorering, en merge eller en opgradering af en afhængighed. En test, der er skrevet til at fejle på defekten og bestå på

rettelsen, gør rettelsen til en varig egenskab ved buildet. Dukker en regression alligevel op, bliver versionshistorikken et søgerum: `git bisect` laver en binær søgning i de commits, der ligger mellem en kendt god og en kendt dårlig version, så et interval på 1.000 commits bliver indsnævret til den, der introducerede fejlen, på omkring ti testkørsler.

Nogle defekter er også sikkerhedsproblemer. MITRE's Common Weakness Enumeration katalogiserer svagheder, som projektet beskriver som tilstande i en software-, firmware-, hardware- eller tjenestekomponent, der under visse omstændigheder kan bidrage til, at sårbarheder opstår. Mange punkter på listen er helt almindelige defekter, som en angriber kan bruge til noget: en manglende grænsekontrol, brugerinput sendt uescapet ind i en SQL-forespørgsel, et autorisationstrin, der springes over på én rute.

Sådan ser det ud i praksis

De defekter, der kostede mest på dette site, var dem, som alle automatiske signaler meldte i orden. En gennemgang afslørede elleve tilgængelighedsdefekter, som de automatiske regler havde godkendt, fordi de alle opstod i den færdigt tegnede side og var usynlige i HTML-koden. To af dem sad i selve kontrollerne. Hvert vindue, layouttjekket åbnede, var 900 pixels højt, så en telefon på tværs i 852 gange 393, med syv pixels mellem to elementer, var aldrig blevet kigget på; og regelmotoren havde kørt 70 af sine 105 regler. Hver defekt kom i et register sammen med den regel, den gav anledning til, så arbejdet lukker en hel klasse af fejltagelser frem for en enkelt forekomst. Da layouttjekket selv var repareret, fejlede det 128 gange på tre sprog.

En anden defekt gav slet ingen symptomer. Sitet har et mørkt tema som standard og et lyst til læsere, hvis system beder om det, og baggrunden, der dækker hele siden, er bygget af tre lag i det mørke tema. Et lille strukturelt tjek, der tæller de lag, hvert tema maler, fandt det lyse tema med kun to allerede ved første kørsel. Overlaget havde manglet, siden det lyse tema blev skrevet. Intet kastede en fejl, hver side blev vist, og en baggrund med ét lag færre ligner et bevidst valg, så i månedsvis var der ingen grund til at kigge efter. Der skulle kun én regel til at rette det; tjekket er det, der holder det rettet.

At holde rettelser på plads er opgaven for repositoriets kontrolmotor, hvor én regel binder hver fejlrettelse til et bevis: en commit, der kalder sig en rettelse, skal medbringe en test, som slår fejl, hvis rettelsen fjernes. Kontrollerne har

desuden 146 test af deres egne, fordi en fejlbehæftet kontrol kan lukke de ændringer igennem, som den skulle afvise, og se ud til at virke.

Hvor det går galt

Tilfældet bliver rettet, og typen står åben. Den indberettede linje bliver rettet, sagen lukkes, og tre søskende af samme fejltagelse venter andre steder i koden. Efter hver rettelse skal du søge efter det mønster, der skabte defekten, og hvor en maskine kan genkende mønsteret, så gør søgningen til et tjek.

En rettet bug kommer igen. En refaktorering et halvt år senere genindfører den gamle opførsel, fordi intet skrev ned, hvad rettelsen beskyttede imod. Skriv testen før rettelsen, se den fejle på den gamle kode, og behold den i testsuiten.

Tjekket bærer selv defekten. Et regelsæt, der uden at sige det kun indlæser en brøkdel af sine regler, en liste af testvinduer, der alle har samme højde, en assertion, der tjekker, hvad koden gør, i stedet for hvad den burde gøre: hver af dem melder succes, mens defekten ligger lige for øjnene af alle. Giv tjekkene deres egne test, og læs i værktøjets eget output, hvilke regler der faktisk kørte.

Dækning træder i stedet for korrekthed. En testsuite kan udføre hver eneste linje og alligevel godkende et forkert resultat, når ingen assertion kigger på udfaldet, og kode, der opfylder sin skrevne specifikation, kan stadig ramme ved siden af et underforstået behov. Brug dækning til at finde kode, som intet tester, og vurder en test efter de defekter, den ville fange.

»Virker som designet« lukker en ægte indberetning. Opførslen svarer til specifikationen, indberetningen afvises, og det defekte arbejdsprodukt var hele tiden specifikationen. Når problemet bor i kravet eller designet, så registrer defekten dér, så rettelsen sker ved kilden.

Læs mere

- Defect — ISTQB's definition fra softwaretest med synonymer og de beslægtede begreber error og failure.
- ISO/IEC 25010 — en oversigt over produktkvalitetsmodellen og underpunkterne for hver egenskab, blandt dem funktionel egnethed og pålidelighed.

- git-bisect — Gits manual for den binære søgning efter den commit, der introducerede en fejl, også automatisk med et testscript.
- About CWE — hvad Common Weakness Enumeration er, hvordan listen over svagheder i software og hardware er organiseret, og hvem der driver den.

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 defekt?« 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/defect/