

TESTERE BØR VÆRE DESTRUKTIVE

Den engelske psykologiforskeren Peter Wason publiserte i 1960 resultatene fra en studie der han ba deltakerne å identifisere regelen han hadde brukt for å lage tallrekker av tre tall. Den eneste måten de kunne skaffe informasjon om korrekt regel var ved å gi Wason testdata og få hans ja eller nei om disse var i henhold til regelen hans eller ikke. På testspråket kan denne situasjonen beskrives som at Wason var deres testorakel og at deltagerne skulle teste om deres algoritme eller program for tallgenerering var korrekt eller ikke. Et eksempel på en tallrekke gitt av Wason var 2-4-6. Tenk selv etter hvilke tallrekker du ville ha testet, dvs spurte Wason om var korrekt eller ikke, før du leser videre.

Den faktiske regelen Wason hadde brukt var "enhver tallrekke av økende tallverdier". Wason observerte at de aller fleste av deltagerne kun spurte om tallrekker som bekreftet hva de trodde var regelen Wason hadde brukt. Nesten ingen spurte om tallrekker som kunne avkrefte regelen de trodde på. Dersom de for eksempel trodde at regelen var "alle tallrekker der tallverdien øker med to", foreslo de f eks 10-12-14, 32-34-36 og 2-4-6. De fikk dermed bekræftende ja av Wason i alle tilfellene, og trodde feilaktig at de hadde funnet korrekt regel. Sjelden eller aldri foreslo de tallrekker som 1-2-3 eller 14-12-10, noe som effektivt ville ha avkrefte at regelen deres var korrekt. Jeg gjorde nylig en slik studie på en klasse av IT-studenter og fant akkurat det samme mønsteret. Etter 10-15 minutter med forslag om tallrekker var det mange som trodde de hadde testet nok til å være rimelig sikker på regelen deres var korrekt, men det var i realiteten ingen klarte det. Som med Wason's deltagere, så var det svært få om noen av tallrekkene som var egnet til å falsifisere reglene de trodde på! Alle var stilt for å bekræfte hva de trodde var riktig regel. Denne tendensen kalles bekræftelsestendens ("confirmation bias") og er funnet i mange ulike situasjoner. Jeg gjorde engang en studie av IT-utviklere og hvordan de brukte data til å bekræfte det de allerede trodde på. Der fant jeg for eksempel at de som trodde på en positive effekt av smidig utvikling så mønstre som bekreftet at smidig utvikling ga gevinster i tilfeldig genererte data, dvs data der det ikke var noen mønstre. De som var mer skeptiske til smidige metoder, så derimot ingen slike mønstre eller mønstre i disfavør av smidig utvikling i de samme tilfeldig genererte data-ene. Vi ser det når vi tror det, er en variant av bekræftelsestendensen.

Det er studier som indikerer at den samme bekræftelsestendensen er typisk for systemutviklere som tester programvare. Teasley, Leventhal, Mynatt og Rohlman (1994) fant for eksempel at testdekningen (test coverage) av programvare var ca. halvparten så god for ekvivalensklasser med lovlige inndata sammenlignet med ekvivalensklasser med ulovlige inndata. Økt erfaring og bedre kravspesifikasjoner reduserte bekræftelsestendensen, men fjernet den ikke. Studier av Calikli og Bener (2010) i flere store IT-organisasjoner indikerte at graden av bekræftelsestendens, blant annet målt som utvikleres prestasjoner på Wason's 2-4-6 test beskrevet ovenfor, i stor grad korrelerte med feilhyppighet i programvaren. Dette

tyder på at et svært viktig kjennetegn ved en god tester er at han/hun i liten grad faller i bekræftelsesfellen, men er god til å lage tester og testdata som er i stand til å avkrefte at programvaren virker. IBM's "the black team" så tidlig tilbake som på 1960-tallet var trolig et team av slike testere. IBM hadde hatt flere kritiske feil i programvaren og måtte gjøre noe med dette. Noe av det de gjorde var å sette sammen et team med testere. Dette teamet utviklet en kultur der hovedoppgaven ikke var å vise at programvaren virket. I stedet skulle de gå inn for å vise at programvaren hadde feil og mangler. Denne gruppen hevdes å ha vært mange ganger så effektiv som andre testgrupper i å finne feil. Fokuset på å være destruktiv som tester er kanskje bare en del av årsaken til at teamet var effektivt, men støtter likevel ideen om at gode tester er destruktive og fokusert på å avkrefte at programvaren virker som den skal.

Graden av avkreftende og destruktiv holdning ved testing varierer, som vist av Calikli og Bener, mellom testere. Det er derfor viktig å velge testere med de rette personeneegenskaper på dette området. Kanskje enda mer viktig er det å skape en testkultur som unngår at testing kun blir en øvelse i å vise at programvaren virker. Forskningen tyder på at vi lett for en uheldig tendens til bekræftelsestendens i testing dersom vi ikke har prosesser og kultur for at testing også i stor grad skal være en destruktiv, avkreftende prosess.