INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2020-08-27

EXAMENSARBETE Mitigation and handling of non-deterministic tests in automatic regression testing

STUDENTER Axel Berglund, Oskar Vateman
HANDLEDARE Per Runeson (LTH), Helena Razdar (Axis Communications)
EXAMINATOR Emelie Engström (LTH)

Minimering och hantering av programvarutester med odeterministiska utfall

POPULÄRVETENSKAPLIG SAMMANFATTNING **Axel Berglund, Oskar Vateman**

Ett programvarutest körs två gånger på raken. Ena gången passerar testet, för att i nästa körning fallera utan att någon ändring har skett däremellan. Det kan låta som en omöjlighet, men faktum är att detta är ett stort problem för programvarutestare. I vårt arbete har vi tagit fram riktlinjer för hur man kan skriva tester för att få ett deterministiskt utfall samt hur man hanterar tester som är odeterministiska.

För att veta om en programvara fungerar som den ska så måste man testa den. Om man dessutom kontinuerligt uppdaterar programvaran så måste man också kontinuerligt testa den för att säkerställa att gammal funktionalitet fortfarande fungerar efter införande av ny funktionalitet. Detta brukar kallas för *regressionstestning* och är en viktig aktivitet vid programvaruutveckling. Regressionstesterna är ofta automatiserade då de exekveras ofta. Ett problem som länge gäckat programvarutestare är att dessa automatiserade regressionstester ibland kan ge olika resultat trots att ingen ändring av programkod eller tester har skett. Ett sådant test med odeterministiskt utfall brukar kallas för ett *flaky test*.

Ett *flaky test* är problematiskt då programvarutestare inte längre kan lita på sina testresultat, vilket i sin tur kan leda till att man inte kan lita på programvaran eller att buggar och felaktigheter smyger sig in i programvaran. Att minimera andelen *flaky tests* skulle därför göra testresultaten mer värdefulla och leda till mer tillförlitlig programvara.

Genom att intervjua testare, utvecklare, och experter inom programvarutestning har vi i vårt examensarbete försökt förstå vad som orsakar *flaky tests* och utifrån detta utvecklat riktlinjer för hur man minimerar andelen *flaky tests* samt hur man hanterar dem.

För hanteringen av *flaky tests* har vi tagit fram en flowchart där man beroende på bland annat testets komplexitet får rekommendationen att antingen fixa testet, ta bort testet, ersätta testet eller ha kvar testet oförändrat.

För att stödja dem som skriver tester tog vi fram åtta riktlinjer. Generellt säger riktlinjerna att man ska göra testerna och testmiljön mindre komplex, t.ex. genom att minska fel som uppstår p.g.a. nätverksanslutning eller att man vill minimera tredjepartsberoenden. Riktlinjerna har valideratats av testare, utvecklare och experter inom programvarutestning men vidare studier bör göras där man testar riktlinjerna i praktiken under en längre period.

Riktlinjerna bör leda till mer tillförlitliga tester inom programvaruutvecklingen, och i förlängningen även mer robust och tillförlitlig programvara.