

Statistisch Testen

Voorwaarden voor succesvolle toepassing ontbreken

Erik van Veenendaal
(Improve Quality Services BV)

Natuurlijk is statistisch testen een uitstekend idee. De vraag is echter of we als testers het probleem kunnen oplossen van onvolwassen IT-organisaties en of alle mogelijkheden die gestructureerd testen ons hede ten dage biedt, al volledig worden benut. Met andere woorden, in hoeverre is deze discussie zinvol? Tenslotte is het van belang te bepalen welke voorwaarden er gelden voor het kunnen toepassen van statistisch testen.

Onvolwassen IT-organisaties

Alvorens te bezien in hoeverre statistisch testen aan de problematiek een oplossing kan bieden, is het belangrijk om eerst een goede analyse te maken van de huidige situatie. Natuurlijk is het zo dat de testkosten toenemen. Een onderzoek van het NGGO uit 1991 geeft aan dat de testinspanning zo'n 20% van de totale ontwikkelingsinspanning bedraagt [1]. Bij huidige projecten is dit percentage vaak 40% of zelfs 50%. Overigens zijn bij veel organisatie en projecten niet zo zeer de (test)kosten, maar veel meer de (test)doorlooptijd kritisch. Time-to-market is steeds vaker van essentieel belang voor het succes van een IT-project.

Het toenemend belang alsmede ook de toegenomen complexiteit van IT-producten hebben helaas niet geleid tot structureel betere IT-organisaties. Nog steeds dient het merendeel van de IT-organisaties te worden gekenmerkt als onvolwassen, en bevinden zij zich in termen van het Capability Maturity Model (CMM) op niveau 1 'Initial' [2]. Recente cijfers laten zien dat slechts 25% van de IT-organisaties zich op CMM niveau 2 'Defined' of hoger bevindt (een overigens stijgend percentage), en dat terwijl inmiddels onomstoten is aangetoond dat hogere volwassenheid leidt tot doorlooptijdreductie en minder fouten. Het probleem dat wordt behandeld is derhalve in eerste aanleg geen testprobleem. De diepere oorzaak ligt bij de IT-organisaties, waarvan het merendeel nog steeds niet in staat is op een goede manier requirements op te stellen en te managen, gedegen projectplannen op te stellen met realistische begrotingen, configuratie-management adequaat uit te voeren, etc. Statistisch testen, of welke testaanpak dan ook, is derhalve niet meer (maar ook niet minder) dan een interessante detectieve maatregel, daar waar we ons zouden moeten focuseren op preventie.

Gestructureerd Testen

In de praktijk worden we natuurlijk veelal geconfronteerd met zogenaamde 'niveau 1' organisaties, en zullen we het testproces zo moeten inrichten dat de productrisico's op een verantwoorde manier worden afgedekt waarbij de testkosten en doorlooptijd binnen bepaalde grenzen moeten blijven. Een gestructureerde testaanpak, zoals de defacto Nederlandse standaard TMap [3], biedt hier een groot aantal mogelijkheden voor, die helaas vaak niet of niet adequaat worden benut. Het startpunt voor een goed testproces is de teststrategie, waarbij op basis van een productrisicoanalyse, de prioriteiten en diepgang van de diverse tests worden bepaald. In veel organisaties wordt tegenwoordig bij een project een teststrategie opgesteld. Men heeft echter vaak moeite om de teststrategie te concretiseren in een gedifferentieerde testaanpak waarbij afhankelijk van het risico meer of minder testgevallen worden opgesteld. In de praktijk wordt gewoon het hele systeem met ongeveer dezelfde diepgang getest. Ook het management, een van de belanghebbende bij het opstellen van de teststrategie, vindt het maken van keuzes waarbij sommige delen van het systeem niet of minder diepgaand worden getest, vaak lastig en soms zelfs onacceptabel. De uitspraak: "Het systeem moet gewoon goed en volledig worden getest", is

helaas nog steeds in gebruik. Als gevolg van de moeilijkheden bij het concretiseren van de teststrategie en het ontbreken van voldoende testawareness bij het management, worden er geen keiharde keuzes gemaakt om delen niet of niet volledig te testen. Achteraf komt dan wel de kritiek dat testen erg veel heeft gekost en te lang heeft geduurd. Met het goed toepassen, in samenspraak met alle belanghebbenden, van de teststrategie is nog heel veel (tijd)winst te behalen!

Het maken van testontwerpen is een activiteit die vooralsnog erg arbeidsintensief is (ondersteunende tools zijn helaas niet of nauwelijks voorhanden). Het heeft echter naast de reeds genoemde voordelen (zie artikel 'Henzen'), minstens nog twee andere belangrijke voordelen. Ten eerste zorgt een goed en gedetailleerd testontwerp er voor dat een fout reproduceerbaar is. Immers alle testacties die zijn uitgevoerd zijn gespecificeerd in het testontwerp. Een ieder die als software ontwikkelaar werkt of heeft gewerkt, weet dat eigenlijk alleen fouten die goed reproduceerbaar zijn op kunnen worden gelost. Ten tweede is het maken van testontwerpen een vorm van statisch testen. Tijdens het opstellen ervan worden allerlei bevindingen gedaan in de uitgangsdokumentatie (functioneel ontwerp, requirements specificatie). Door deze bevindingen serieus te behandelen, wordt in een vroegtijdig stadium reeds een groot aantal bevindingen gedaan die er toe zullen leiden tot er minder dure bevindingen (zowel qua tijd als geld) worden gedaan tijdens de testuitvoeringsfase. Een gedegen georganiseerde en uitgevoerde fase testontwerp verdient zich op deze wijze al grotendeels of zelfs volledig terug! Tenslotte is het testontwerp geen activiteit op het kritieke pad (mits tijdig uitgevoerd) en heeft deze derhalve geen negatieve invloed op de time-to-market van het product. Iets wat zoals eerder is aangegeven, vaak veel belangrijker is dan de eenmalige ontwikkelkosten.

Statistisch testen

Is er nog een rol voor statistisch testen? In het Testing Maturity Model [4],[5] komt statistisch testen nadrukkelijk aan bod op niveau 5 'Optimisation' als onderdeel van het procesgebied 'Quality Control'. Quality Control heeft zijn oorsprong in industriële productieprocessen waar a-selecte steekproeven, metriecken, oorzaakanalyse en acceptatie steekproeven technieken zijn om de kwaliteit van de geproduceerde eindproducten vast te stellen en te borgen [6]. Statistisch testen is niet slechts het willekeurig nemen van een steekproef, maar het op basis van gebruiksprofielen een representatieve testset samenstellen om vervolgens door middel van metingen een uitspraak te kunnen over de kwaliteit van het totale product. Een essentieel onderdeel van statistiek is uiteraard dat door middel van de steekproef een betrouwbare uitspraak wordt gedaan over de totale populatie. Dit toegepast op software producten betekent dat slechts een deel wordt getest en vervolgens een uitspraak wordt gedaan over de kwaliteit van het totale systeem. Deze werkwijze veronderstelt dat het product een min of meer homogeen kwaliteitsniveau heeft; overal dezelfde complexiteit, codeer standards zijn consequent toegepast, de ontwerpen zijn allen geïnspecteerd etc. Verantwoord statistisch testen vereist een gedefinieerd en reproduceerbaar ontwikkelproces (niet voor niets komt het pas op niveau 5 van het TMM aan bod), met andere woorden een volwassen IT-organisatie en dat is nu net waar het nog vaak aan schort. Statistisch testen (en quality control) zijn uitstekende werkwijzen, echter aan de voorwaarden voor een succesvolle toepassing wordt nog maar zelden voldaan. Kunnen we dan voorlopig niets doen met statistisch testen? Een aantal concepten is zeer wel bruikbaar op lagere volwassenheidsniveaus en kunnen uitstekend *complementair* zijn ten opzichte van het 'traditionele' gestructureerd testen. Met name de ideeën ten aanzien van testen op basis van gebruiksprofielen kunnen een bijdrage leveren aan onder andere het acceptatietesten en performance testen. Ook het opbouwen van statistische metriecken zoals het 'aantal fouten per testuur', en het gebruik daarvan als acceptatiecriterium is een welkome verrijking van de meer traditionele acceptatiecriteria zoals 'het aantal openstaande fouten'. Een gedegen studie van statistisch testen door de testgemeenschap is iets wat een bijdrage aan het verbeteren van het

hedendaagse testproces kan leveren. Delen van ervan zullen toepasbaar zijn binnen gestructureerd testen. Voorlopig zijn we echter nog ver verwijderd van de volledig (en alleen) statistisch testen, het zal nog wel even duren voordat we onder een XRAY-scanner gaan liggen waarbij tijdens de testfase alleen met een steekproef de kwaliteit is bepaald.

Erik van Veenendaal is een internationaal erkend expert op het gebied van software kwaliteit en testen (o.a. co-auteur van 'Testen volgens TMap' en keynote tijdens de, dit jaar in Amsterdam te houden, EuroSTAR testconferentie). Hij is directeur van Improve Quality Services BV en als universitair docent verbonden aan de TU-Eindhoven. (eve@improveqs.nl)

[1] NGGO (1991), *Over het testen van informatiesystemen*, Uitgeverij Tutein Nolthenius, 's Hertogenbosch

[2] Carnegie Mellon University – Software Engineering Institute (1995), *The Capability Maturity Model*, Addison-Wesley

[3] Pol, M, E. van Veenendaal en R. Teunissen (1999), *Testen volgens TMap*, 2^e druk, Uitgeverij Tutein Nothenius, 's Hertogenbosch

[4] Burstein, I. (2002), *Practical Software Testing*, Springer Professional Computing

[5] Veenendaal, E. van (2002), *The Testing Practitioner*, Uitgeverij Tutein Nothenius, 's Hertogenbosch

[6] Juran, J.M. and F.M. Gryna (1970), *Quality Planning and Analysis*, McGraw-Hill Book Company

Testen als statistisch proces

Rob Henzen

Inleiding

Testactiviteiten leggen in toenemende mate beslag op projectbudgetten, terwijl veelal onduidelijk is of de kwaliteit van het geleverde testwerk wel gelijke tred houdt met deze kostenontwikkeling. De oorzaak van toenemende testkosten is algemeen bekend: bedrijven zijn in toenemende mate afhankelijk van de in gebruik zijnde informatiesystemen. Falende geautomatiseerde systemen vormen derhalve een groot risico voor de continuïteit van de bedrijfsvoering. Om dit risico te kunnen inschatten en waar mogelijk te verkleinen worden tests uitgevoerd op nieuwe of aangepaste informatiesystemen. Maar met de toenemende afhankelijkheid die zij hebben voelen bedrijven zich genoodzaakt deze systemen steeds intensiever en uitgebreider te testen. Met andere woorden: er worden steeds hogere eisen gesteld aan de kwaliteit van de gebruikte testgevallen.

In sommige gevallen poogt men deze kwaliteitsverhoging te bereiken door simpelweg steeds meer testgevallen te gebruiken, in het geval van releasematig onderhoud bijvoorbeeld door bij elke release een volledige regressietest uit te voeren. Een andere benadering bestaat er uit om steeds verfijndere methoden en technieken toe te passen teneinde de ‘juiste’ testgevallen te kunnen identificeren. Onder juiste testgevallen worden dan testgevallen verstaan waarmee eventuele bedrijfsrisico's beter kunnen worden ingeschat en zo mogelijk kunnen worden geminimaliseerd. Testsets worden in deze benadering dus steeds fijnmaziger van opzet. Het is echter maar de vraag of deze strategie van steeds grotere of fijnmaziger testsets opstellen ook inderdaad het gewenste effect sorteert.

Wellicht is het een goed idee om een geheel andere benadering te kiezen. en ons af te vragen of er niet veel meer te winnen valt bij het ‘op goed geluk’ uitkiezen van testgevallen en het toeval (ofwel de statistiek) haar werk te laten doen?

De praktijk

Hoewel er verschillende manieren zijn waarop een testtraject kan worden ingericht, hebben veel van deze manieren een aantal gemeenschappelijke kenmerken. Zo is het goed gebruik om voor een testtraject een testplan op te stellen; een plan van aanpak, vergelijkbaar met een projectplan. In het testplan worden zaken behandeld als op te leveren producten, uit te voeren activiteiten, uitgangspunten en randvoorwaarden, en een planning.

Verder wordt veel aandacht besteed aan het opstellen van testgevallen en het vastleggen hiervan in zogenaamde testontwerpen. Dit laatste vooral met als doel de uit te voeren tests herhaalbaar, controleerbaar en overdraagbaar te maken. Alvorens te kunnen overgaan tot het maken van een testontwerp is het overigens van belang eerst antwoord te hebben op de vraag ‘hoeveel testgevallen moeten er gemaakt worden?’. En niet minder belangrijk: ‘welke testgevallen moeten er gemaakt worden?’. Met andere woorden: wat moeten we wel testen en wat niet? Het is immers evident dat het bij een voldoende groot informatiesysteem onmogelijk is om alle mogelijke testgevallen uit te voeren. De tijd en het budget hiervoor ontbreken gewoonweg, mede gezien de hierboven geschetste ontwikkelingen rond intensiteit en doorloop van testtrajecten. Hierdoor ziet elke testontwerper zich genoodzaakt om een keuze te maken uit de vele mogelijke testgevallen die te bedenken zijn.

De afgelopen jaren is veel tijd en energie gestoken in het vaststellen van werkwijzen en criteria die zouden moeten leiden tot een verantwoorde keuze van testgevallen. Een veel gebruikte methode om te kunnen vaststellen hoe de aard en spreiding van de uit te voeren testgevallen moet zijn is het uitvoeren van een risicoanalyse. Hierbij wordt voor de diverse onderscheiden systeemcomponenten een inschatting gemaakt van het (bedrijfs)risico dat wordt gelopen

wanneer de betreffende component niet of niet goed zou functioneren. Vervolgens wordt de testinspanning (i.e. het aantal op te stellen en uit te voeren testgevallen per systeemcomponent) verdeeld op basis van de uitkomst van deze risicoanalyse, veelal volgens het principe: hoe groter het risico, hoe meer testgevallen er nodig zijn.

Het vervolgens daadwerkelijk opstellen van de testgevallen gebeurt over het algemeen met gebruikmaking van testspecificatietechnieken; technieken om op gestructureerde wijze testgevallen af te leiden uit een document. Voorbeelden van testspecificatietechnieken zijn beslissingstabellen, algoritmetesten en dataflowtesten. In de meeste gevallen komt het toepassen van deze technieken neer op het vertalen van de uitgangsdokumentatie (ofwel de testbasis) naar een meer gestructureerde vorm (algoritme, beslissingstabel) waaruit de mogelijke testgevallen eenvoudig(er) zijn af te lezen. Vervolgens worden de testgevallen uitgeschreven in een testontwerp. De vorm waarin dit gebeurt volgt in veel gevallen de klassieke driedeling van 'uitgangssituatie' (wat moet je hebben om de test uit te kunnen voeren), de 'actie' (wat moet je doen), en de 'uitvoerspelling' (wat is het verwachte resultaat).

De voordelen van bovengeschetste werkwijze zijn evident: het zwaartepunt van de test ligt daar waar de meeste risico's worden gelopen. En de te gebruiken testgevallen worden op dusdanige manier opgesteld dat controleerbaar is hoe de testontwerper aan zijn of haar testgevallen is gekomen. Verder is de test door het vastleggen van de testgevallen in een testontwerp herhaalbaar, onderhoudbaar en overdraagbaar gemaakt.

Er kleeft aan deze methode echter ook een groot nadeel; zij is erg arbeidsintensief. Vooral de vertaling van de uitgangsdokumentatie naar een meer gestructureerde vorm en het vervolgens uitschrijven van de testgevallen kost in het algemeen erg veel tijd. Bij een project van enige omvang is het niet ongebruikelijk dat een aantal testontwerpers tenminste enkele weken, soms zelfs maanden, bezig zijn met het opstellen van de testontwerpen. Gevolg van deze werkwijze is dan ook dat testactiviteiten regelmatig zo'n 40 tot 50 procent van het totale projectbudget opsouperen. Hier komt uiteraard bij dat deze werkwijze een zware wissel trekt op de totale doorloop van een project.

Alternatief

Er is echter een alternatief voorhanden. Het testen van een geautomatiseerd systeem is namelijk op te vatten als het nemen van een steekproef uit de populatie van het totaal aantal mogelijke testgevallen. Met als doel het afleiden van de kwaliteit van het gehele systeem op basis van de resultaten van de steekproef. Maar als testen niets meer of minder is dan het nemen van een steekproef, moeten ook de regels die gelden voor het nemen van een steekproef worden geëerbiedigd! En door consequent deze regels te volgen kan vervolgens enorm veel tijd (en geld) worden bespaard.

Testgevallen dienen in dat geval namelijk *volkomen willekeurig* te worden gekozen. Het maken van een testontwerp is dan ook weinig zinvol, en zelfs af te raden. Bij het maken van een testontwerp komen de testgevallen immers verre van willekeurig tot stand; de testontwerper bepaalt, al dan niet op basis van vastgelegde criteria, welke testgevallen wel en welke niet opgenomen zullen worden in de testset. Dit soort menselijke keuzes dienen bij het nemen van een steekproef nu juist zoveel mogelijk vermeden te worden. In het ideale geval zouden de testgevallen zelfs door een 'random generator' moeten worden vastgesteld.

Met deze methode is een enorme tijdwinst te boeken op het testtraject en derhalve op de totale projectdoorloop. En zonder dat wordt ingeleverd op de kwaliteit van de uitgevoerde test. Integendeel: er wordt gewerkt volgens de regels die gelden voor het samenstellen van een goede steekproef, en de resultaten zijn dienovereenkomstig betrouwbaarder.

Bijkomend voordeel van de statistische benadering van testen is dat het minimum aantal testgevallen dat nodig is voor het bereiken van de gewenste nauwkeurigheid (van de kwaliteitsvoorspelling) op eenvoudige manier eenduidig kan worden vastgesteld.

Hierdoor vervalt voor een groot deel de noodzaak tot het uitvoeren van een diepgaande risicoanalyse, die zoals we hebben gezien immers voornamelijk tot doel heeft om vast te stellen hoeveel testgevallen er per systeemonderdeel moeten worden uitgevoerd. Ook dit levert een flinke besparing op de totaal benodigde tijd die voor een testtraject nodig is.

Conclusie

Gezien het steeds grotere beslag dat testen legt op het budget en de doorloop van automatiseringsprojecten zouden we ons wellicht moeten afvragen of we met het streven naar steeds betere en uitgebreidere testsets wel op de goede weg zijn.

Is in de huidige situatie het te bereiken (test)doel nog wel in balans met de daarmee gepaard gaande kosten? En is er misschien een alternatief, dat minder arbeidsintensief is, maar toch een betrouwbaar beeld oplevert van de kwaliteit van het geteste product?

Zoals hierboven geschetst zou een alternatief kunnen zijn het uitvoeren van een test op een informatiesysteem op te vatten als het nemen van een steekproef, waarbij dan natuurlijk wel de regels die gelden voor het nemen van zo'n steekproef consequent gevolgd moeten worden. Dit kan een positief effect hebben op de kosten en de doorloop van een testtraject, en daarmee op het gehele project, zonder dat noemenswaardige afbreuk wordt gedaan aan de kwaliteit van de uitgevoerde test.

Alleen al vanwege dit laatste zou het de moeite waard kunnen zijn eens nader te kijken naar 'testen als statistische activiteit'.

De auteur

Rob Henzen werkt als (test)consultant bij Refis *system reliability engineering* te Bilthoven (www.refis.nl). Hij houdt zich voornamelijk bezig met betrouwbaarheidsanalyses van informatiesystemen, het opzetten en begeleiden van testtrajecten, en het uitvoeren van audits op test- en ontwikkeltrajecten.