

Testen met

Regelmatig klagen opdrachtgevers, ontwerpers, ontwikkelaars en zelfs testers dat het gebruik van testontwerptechnieken moeilijk en tijdrovend is. Liever werken ze uit de losse pols. Dat is jammer omdat het gebruik van deze technieken de enige mogelijkheid biedt om de afgesproken teststrategie aantoonbaar waar te maken, en vaak wordt het testen nog efficiënter ook.

Ontwerptechnieken maken teststrategie!

Het 'bindmiddel' tussen alle stakeholders

Een teststrategie wordt met alle stakeholders opgesteld. Dus onder andere met de opdrachtgever, gebruikers, architecten, ontwerpers, ontwikkelaars, testers en beheerders. In de teststrategie wordt vastgelegd welke risico's het gevolg kunnen zijn van een nieuw of gewijzigd IT-systeem. De risico's worden geclassificeerd, waarna maatregelen worden getroffen om de risico's te beperken. Testen is een van deze maatregelen en hierbij heeft het toewijzen van testontwerptechnieken als doel het concretiseren van de goedgekeurde teststrategie. Bij hoge risico's zijn meer en betere testgevallen nodig dan bij lage risico's. En als er helemaal geen risico is? Dan adviseren wij om het betreffende deel van het IT-systeem niet te bouwen. Want als niemand er last van heeft als het niet werkt, dan heeft niemand het nodig, toch?

Het kiezen van een bij het risiconiveau passende ontwerptechniek is lastig. In diverse testboeken is een groot aantal technieken toegelicht, TMap NEXT[®] beschrijft er 22, in Zweden is iemand afgestudeerd op ruim 70 technieken. De keuze is, naast het te testen kenmerk en de gekozen testzwaarte, sterk afhankelijk van aspecten als testbasis, kennis en kunde van de testers en de arbeidsintensiviteit. Dit artikel biedt hulpmiddelen om één of meer passende technieken te kiezen. Voor meer informatie verwijzen wij naar het boek TMap NEXT[®] Business Driven Test Management.

De testrol in dit artikel beperkt zich niet tot de 'professionele tester'. Ook de andere genoemde stakeholders vervullen op een zeker moment een testrol in het project. En juist voor deze 'niet-professionele' testers is het van belang een techniek te kiezen die ze begrijpen, kunnen toepassen en waarmee de

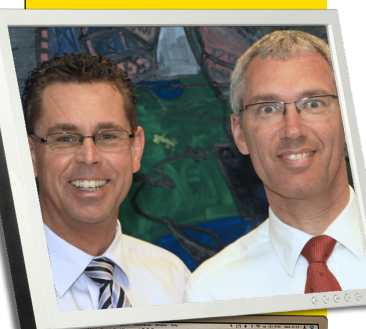
teststrategie kan worden waargemaakt. Overal waar in dit artikel "tester" staat, wordt dan ook een stakeholder bedoeld in de rol van "tester".

Moeilijk en tijdrovend?

Hoe komt het toch dat testers het gebruik van testontwerptechnieken vaak als moeilijk en tijdrovend zien? Het lijkt erop dat hier het gezegde, "onbekend maakt onbemind", opgaat.

Er zijn veel technieken, sommige zijn abstract of lijken erg wiskundig, daarom worden ze vaak als moeilijk ervaren. Om te voorkomen dat de tester 'door de bomen het bos niet ziet', is het dan ook goed om niet alle testontwerptechnieken te leren. In de praktijk blijkt dat als enkele (3 of 4) technieken al genoeg zijn voor succes. Als ze dan goed worden uitgelegd, met voorbeelden uit de directe werkomgeving, vindt de tester de technieken 'opeens' niet meer moeilijk en ziet de voordelen van het gebruik.

Testers die roepen dat een testontwerptechniek tijdrovend is, zetten deze technieken vaak 'verkeerd' in. Het doel is niet dat de tester alle mogelijke testgevallen ontwerpt, maar kiest voor een bepaalde techniek die past in de gekozen teststrategie, waarmee zoveel testgevallen worden gemaakt als past bij het risico van een bepaald onderdeel of aspect van het systeem. Als je één beslispunt hebt met 4 condities dan kun je dat afdekken met 2 testgevallen, of met 5 of met 16, afhankelijk van het risico en de bijbehorende testontwerptechniek. In de praktijk blijkt dat regelmatig een verkeerde keuze wordt gemaakt, met als gevolg dat er onnodig veel testgevallen worden ontworpen. Anders gezegd: Testontwerptechnieken maken of breken de teststrategie!



Leo van der Aalst en
Rik Marselis

De juiste techniek kiezen

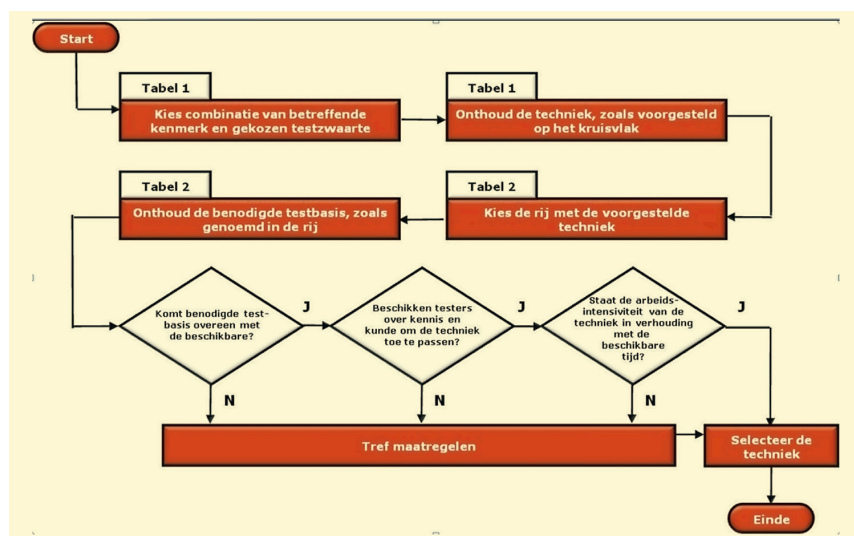
De keuze van een testontwerptechniek is niet altijd even eenvoudig. De tester moet met een groot aantal variabelen rekening houden. Bijvoorbeeld met “Wat moet getest worden?” Sommige testontwerptechnieken zijn vooral geschikt voor het testen van de interactie (schermen, rapporten, online) tussen systeem en gebruiker. Andere zijn meer geschikt voor het testen van de relatie tussen de administratieve organisatie en het systeem, voor het testen van performance of beveiliging, of voor het testen van complexe verwerkingen (berekeningen). Van het risico hangt af hoe zwaar een bepaald deel van het te testen systeem moet worden getest. Om aan de gekozen testzwaarte te kunnen voldoen moet een passende testontwerptechniek worden gekozen. Maar ook de testbasis (bijvoorbeeld de ontwerpdocumenten) maken veel verschil. Met requirements op het spreekwoordelijke bierviltje is weinig aan te vangen en zal de tester bijvoorbeeld eindgebruikers moeten interviewen, maar is het functioneel ontwerp (als andere uiterste) in pseudocode geschreven dan zijn de testgevallen heel rechttoe-rechtaan gestructureerd af te leiden.

Het stroomschema in figuur 1 (“Selectieschema technieken”) helpt bij het kiezen van de passende techniek. Hierbij hebben wij als voorbeeld de tabellen 1 en 2 gegeven, voor de volledige tabellen verwijzen we naar het eerder genoemde boek.

Onderaan figuur 1 staat de actie “Tref maatregelen”, hierbij kunt u onder andere denken aan het aanpassen van de testbasis door architecten en ontwerpers zodat een techniek toegepast kan worden. Want als bijvoorbeeld de procescyclustest van grote toegevoegde waarde zou zijn is er wel een processchema als basis nodig. Of wanneer de kennis en kunde van de tester niet voldoende zijn (geen ervaring met een bepaalde techniek) dan kun je ofwel de tester opleiden of kiezen voor een techniek die beter bij zijn ervaring past. En het komt ook voor dat er onvoldoende tijd beschikbaar is om een bepaalde techniek toe te passen. Dan kan de teststrategie herzien worden (dit betekent dat de opdrachtgever meer risico moet accepteren, want met minder testen blijft meer risico over). Of de opdrachtgever kan natuurlijk meer tijd beschikbaar stellen.

Resultaat

De techniek die via bovenstaande werkwijze is verkregen, is een voorstel. De tester en/of testmanager kan natuurlijk redenen hebben om in overleg met betrokkenen een andere techniek te kiezen. Het selectieschema is hierbij een hulpmiddel. Soms (zoals in de luchtvaartwereld of farmaceutische industrie) wordt de te gebruiken techniek door regelgeving aan de testorganisatie opgelegd. Uiteindelijk levert de tester bij toepassen van de passende testontwerptechniek die testgevallen op,



Selectieschema technieken.

Kenmerken	Testzwaarte		
	Lichte dekking	Gemiddelde dekking	Zware dekking
Beheerbaarheid	Afvinklijst PCT-testmaat 1 UCT-afvinklijst EG	DCT-equivalentieklassen PCT-testmaat 2 ET	DCT-pairwise testing PCT-testmaat 3
Beveiliging	Afvinklijst EG	DCT-equivalentieklassen SEM-modified condition/decision coverage ET	DCT-pairwise testing Penetratietest
Bruikbaarheid	UCT-afvinklijst	PCT-testmaat 2 UCT-paden	RLT-operational/load profiles UCT-beslispunten

Voorgestelde techniek bij een bepaalde combinatie van kenmerk en gekozen testzwaarte.

Techniek	Testbasis							
	Alle soorten testbasis	Afzonderlijke condities of beslistabellen zonder structuur	Gestructureerde functionele specificaties (pseudocode)	CRUD-matrix, detailintegriteitsregels	Gestructureerde beschrijving van bedrijfs- of werkprocessen	Operational profiles load profiles	In- en uitvoerspecificaties, business rules	In en uitvoerspecificaties, attributbeschrijvingen
Afvinklijst	x	x					x	x
Beslistabeltest (BTT)	x	x	x					
Data combinatorietest (DCT)	x	x	x	x				
Error guessing (EG)	x	x	x	x	x	x	x	x
Exploratory testing (ET)	x		x			x	x	
Elementaire vergelijkendetest (EVT)		x	x					
Functionele tests		x					x	

De benodigde testbasis bij een voorgestelde techniek.

die rechtstreeks aansluiten bij de gekozen teststrategie. Hiermee krijgen de diverse betrokkenen zoals de opdrachtgever en de architect aantoonbaar de afgesproken afdekking van de risico's die met het nieuwe systeem samenhangen. Een goed uitgewerkte teststrategie is dan ook een uitstekend ‘bindmiddel’ tussen alle stakeholders.

Tevens is onze ervaring dat de testers plezier hebben in het testen omdat ze precies genoeg testen en hun energie gericht toepassen.

Leo van der Aalst is research & development manager bij Sogeti en lector Software Quality and Testing aan de Fontys Hogeschool. Hij is te bereiken via leo.vander.aalst@sogeti.nl.

Rik Marselis is senior testconsultant bij Sogeti en bestuurslid van de vereniging TestNet en de stichting BNTQB. Hij is te bereiken via [rik.marselis@sogeti.nl](mailto:marselis@sogeti.nl).