



Dennis Stam MSC CISA
EMITA
is director binnen de
Technology-groep van KPMG
Advisory.

De organisatie van het testen in relatie tot de gebruiker

Vijftig jaar later

Vijftig jaar geleden was Han Urbanus een van de oprichters van een nieuw vakblad op het gebied van het beheren, beheersen, organiseren en verbeteren van informatievoorziening: *Compact*. In 1974 schreef hij een artikel waarin hij uitweidde over de rol van de eindgebruiker bij het testen van een nieuw (of gewijzigd) informatiesysteem en hoe het testen georganiseerd zou kunnen worden. Nu, een halve eeuw later, kijken we terug op het artikel: heeft de inhoud van het artikel de tand des tijds doorstaan?

INLEIDING

In 1974 schreef Han Urbanus een artikel voor *Compact*, een nieuw vakblad op het gebied van het beheren, beheersen, organiseren en verbeteren van informatievoorziening waar hij een van de oprichters van was. In het artikel ([Urba74]) weidde Urbanus uit over de rol van de eindgebruiker bij het testen van een nieuw (of gewijzigd) informatiesysteem en hoe het testen georganiseerd zou kunnen worden.

In dit artikel wordt eerst een korte samenvatting gegeven van de inzichten die Urbanus in 1974 al had ten aanzien van het organiseren van gebruikerstesten. Daarna wordt stilgestaan bij de wijze waarop testen tegenwoordig plaatsvindt.

TESTEN IN 1974

Al in 1974 was er sprake van een revolutie in de wereld van softwareontwikkeling. Steeds vaker werd – terecht – gewezen op de verantwoordelijkheid die de business heeft ten aanzien van de informatie in en vanuit informatiesystemen. Eindgebruikers zijn daarom ook cruciale stakeholders die hun fiat moeten geven voor (in hun opdracht ontwikkelde) nieuwe systemen of wijzigingen aan bestaande systemen. De vraag was echter: op welke wijze kan een eindgebruiker vaststellen dat het systeem conform de door hem geformuleerde eisen is ontwikkeld?

Urbanus stelde voor het testen van informatiesystemen als een aparte fase binnen de systeemontwikkeling te beschouwen. (Ook) in die tijd was het niet ongebruikelijk dat het testen van een groot en complex systeem minstens net zo veel tijd in beslag nam als het ontwerpen en coderen daarvan.

Behalve dat het testproces als een aparte fase moest worden beschouwd, gaf Urbanus ook al aan dat gebruikers gedurende de gehele ontwikkelcyclus erbij betrokken zouden moeten worden. Tijdens het vooronderzoek zouden de criteria voor systeemacceptatie moeten worden vastgesteld. Deze criteria vormen de basis voor het systeemontwerp en daarmee ook voor de test- en (data) conversieactiviteiten. Gebruikers zijn echter doorgaans niet bekend met testen en bezitten niet de vaardigheden om een testset samen te stellen. Daarom zouden gebruikers al vanaf aanvang bij het proces moeten worden betrokken om te ‘trainen’ op het samenstellen van testen, zodat zij een zo groot mogelijke waarde leveren voor de systeem- en acceptatietesten.

Niet enkel werd er onderscheid gemaakt tussen systeem- en acceptatietesten, Urbanus stelde in 1974 al een (sub-)fasering voor systeemtesten voor:

- testen van individuele modules;
- testen van individuele programma's;
- testen van programmaserieën (ketens);
- testen per deelproject;
- testen van het gehele project.

Door kleine onderdelen eerst te testen en – bij een goed resultaat – daarna op te schalen naar een groter geheel en dat te testen, zorgt deze subfasering ervoor dat de complexiteit afneemt. Hierdoor kan zorgvuldiger worden getest en neemt het zoeken naar een speld in de (IT-)hooiberg af. Helaas, zo constateerde Urbanus, wordt het testen van modules steeds minder vaak uitgevoerd door ongeduld (men wil zo snel mogelijk het volledige programma kunnen toetsen) en door besparingen.

Urbanus eindigde met een reeks richtlijnen en de oproep om naar een meer georganiseerde en gedisciplineerde aanpak van het testen toe te werken:

- Formuleer maatstaven met betrekking tot de acceptatie van het programma.
- Stel een testplan op.
- Ontwikkel testgevallen.
- Beschrijf de testgegevens en het speciale doel daarvan.
- Standaardiseren de job control van de testwerkzaamheden.
- Werken vanuit een source library om herhaling van zetten mogelijk te maken.
- Bepaal vooraf de gezochte resultaten van testen.
- Ontwikkel utilities, zoals voor outputvergelijking.
- Test methodisch en houd daarbij goed in de gaten wat wel en niet getest is.
- Houd een testlog bij.
- Laat de gegevens voor systeemtesten vervaardigen door gebruikers.

DE ONTWIKKELING VAN HET PROFESSIONEEL TESTEN

Urbanus (b)lijkt een pionier op het gebied van testmanagement. In de testgemeenschap staat Glenford Myers bekend als de eerste die, in 1979 (vijf jaar na het artikel van Urbanus), in zijn boek *The Art of Software Testing*, een definitie gaf van ‘testing’:

‘(...) the process of executing a program with the intent of finding errors’.

Uit de snelle ontwikkeling van informatiesystemen in de jaren tachtig bleek echter dat de groeiende complexi-

teit vroeg om meer dan slechts 'het vinden van fouten'. Er werd een meer gestructureerde aanpak gezocht.

In 1995 werd het eerste boek rond de, op Nederlandse grond ontwikkelde, TMap-testaanpak gepubliceerd. *Testen volgens TMap* richtte zich op het definiëren van testen als een gestructureerd proces en introduceerde daarbij ook de volgende definitie:

'Testing is een proces van plannen, voorbereiden en meten, gericht op het vaststellen van de kenmerken van een informatiesysteem en het aantonen van het verschil tussen de actuele en de gewenste status.'

TMap bracht de (IT-)wereld een marktstandaard voor testen en testmanagement. Enkele jaren daarna, in 1998, werd de International Software Testing Qualifications Board (ISTQB) opgericht, een internationale organisatie die gestandaardiseerde opleidingen met toetsen en certificaten voor softwaretesters aanbiedt.

Maar in een wereld waarin IT steeds belangrijker – zo niet onontbeerlijk – werd voor organisaties, was de evolutie van de testaanpak noodzakelijk. Nieuwe inzichten in testen en systeemontwikkeling, waaronder de opkomst van iteratieve en incrementele ontwikkelmethodieken en Agile in het bijzonder, hebben geleid tot nieuwe versies van TMap. Deze nieuwere versies brachten een afweging tussen kosten en baten (onder andere door het toepassen van risk-based testing) en sloten aan op de vernieuwingen op systeemontwikkelingsgebied.

EEN RETROSPECT OP HET OORSPRONKELIJKE ARTIKEL

Veel van de door Urbanus geponeerde stellingen zijn onderdeel geworden van de hedendaagse (professionele) testaanpak zoals door TMap of ISTQB voorgesteld. Testen is een aparte fase geworden, er wordt onderscheid gemaakt tussen systeem- en acceptatietesten, er worden *vooraf* testplannen gemaakt, gebruikers *worden* betrokken bij het testen, et cetera. Kortweg: testen is een professie geworden waarbij een gestructureerde aanpak wordt gehanteerd. Toch zijn er nuances tussen de wijze waarop dat nu gebeurt en de wijze waarop dat gebeurde in de wereld waarin Urbanus zijn artikel schreef.

Het onderscheid in de testfasering is geadopteerd. Tegenwoordig wordt veelal onderscheid gemaakt tussen ontwikkeltesten (door de ontwikkelaar), systeemtesten (door testspecialisten) en acceptatietesten (door acceptanten). Binnen deze testen wordt ook gewerkt volgens het principe dat Urbanus voorstelde: begin met het testen van kleine modules en bouw langzaam op naar

het testen van integraties, programma's en uiteindelijk ketens. Waar Urbanus echter nog een terugval waarnam in de toepassing van ontwikkeltesten, onderkennen veel ontwikkelaars tegenwoordig het belang van goede unittesten en streven naar het (geautomatiseerd) uitvoeren van deze testen met een dekking van ten minste 80 procent van de aanwezige programmacode. Dit helpt om fouten vroegtijdig op te sporen en daarmee hoge kosten voor herstel in latere fasen van de systeemontwikkeling te voorkomen.

Vergelijkbaar met de opkomst van het unittesten heeft de ontwikkeling van 'continuous integration' (CI) en 'continuous delivery' (CD) gezorgd voor enerzijds structuur in de wijze waarop softwarecode wordt beheerd (met een versiemanagementsysteem) en anderzijds een herhaalbaar proces. Wijzigingen worden niet langer direct aangebracht in het informatiesysteem, maar de broncode wordt bijgehouden in een versiemanagementsysteem zoals Git of SVN. Vanuit dit systeem kan een nieuwe versie van de software worden opgesteld en gecompileerd en worden uitgerold naar een (test) omgeving. Hierdoor heeft de tester meer duidelijkheid over (de versie van) het testobject en kunnen nieuwere versies – met bijvoorbeeld de meest recente fixes – eenvoudiger beschikbaar worden gesteld voor testen. Daarnaast levert deze opzet een basis voor het geautomatiseerd uitvoeren van unit- en (een deel van) systeemtesten, waardoor er ook al meer zekerheid is over de kwaliteit van het testobject wanneer de tester aan de slag gaat met een nieuwe versie.

Ook de rol van de gebruikersorganisatie is geformaliseerd: in vrijwel ieder testprogramma wordt, naast de inzet van testprofessionals, een vertegenwoordiging vanuit de organisatie gevraagd om mee te testen. In sommige gevallen worden de eindgebruikers zelfs gevraagd om al tijdens de ontwikkeling van het systeem mee te

Urbanus (b)lijkt een pionier op het gebied van testmanagement

werken aan het toetsen van het ontwerp en het uitvoeren van testen tijdens de systeemtestfase. Bij de uitvoering van de acceptatietesten wordt tegenwoordig echter onderscheid gemaakt tussen verschillende acceptatietestsoorten, zoals de functionele acceptatietest (FAT), gebruikersacceptatietest (GAT), productieacceptatietest (PAT) en beheeracceptatietest (BAT). Gebruikers zijn daarbij de uitvoerenden binnen de GAT (en beheerders, tevens te beschouwen als een speciale gebruikersgroep, binnen de BAT), waarbij het besef dat gebruikers vooral de subject matter expertise moeten inbrengen prevalent. Om die reden worden gebruikers dan ook vaak ondersteund door testspecialisten bij het managen van de testvoorbereiding en -uitvoering, het definiëren van de testset en het vervaardigen van de testdata.

De groeiende complexiteit van informatiesystemen heeft een vergelijkbare complexiteit in de gebruikte data gebracht. Waar in 1974 nog simpelweg gesproken werd over het opstellen van testdata door gebruikers, vergen moderne systemen een veelvoud aan inzet om tot een goede set testdata te komen. Testers hebben immers behoefte aan testdata gericht op (nog) niet voorkomende situaties, waarvoor testdata gefabriceerd moeten worden, en daarnaast aan testdata die zo veel mogelijk representatief zijn voor de real-life situaties waar het informatiesysteem mee zal moeten omgaan. De opkomst van privacywetgeving zoals AVG bemoeilijkt dit laatste de afgelopen jaren: gegevens die in de productieomgeving bestaan, bevatten veelal privacygevoelige data en zijn niet verzameld met het doel om te kunnen testen, en mogen om die reden niet gebruikt worden voor testen. Testdatamanagement is daarmee een opkomende specialisatie gericht op het vervaardigen van data voor niet-productiedoeleinden en betreft het anonimiseren, subsetten, masken, scrubben, 'profielen' en 'agen' van transacties en het 'provisionen' van data.

Naast een groeiende behoefte aan het managen van testdata, vragen de frequentie van opleveren en de toenemende complexiteit van informatiesystemen om een steeds grotere mate van testautomatisering en inzet van tooling. De door Urbanus genoemde 'utilities' zijn dan ook niet langer meer kleine hulpprogramma's voor outputvergelijking of het maken van snapshots, maar volledig geautomatiseerde testsets, emulatie van softwarecomponenten (service virtualization) en tools om testen op meerdere platforms tegelijk uit te voeren (bijvoorbeeld voor het testen van mobiele apps of webapplicaties).

CONCLUSIE

Al met al heeft de wereld van het testen niet stilgestaan in de afgelopen vijftig jaar. Urbanus was met recht een

visionair op dit gebied, getuige het veelvoud aan richtlijnen die hij in 1974 al opstelde en die uiteindelijk in de jaren negentig deel zijn gaan uitmaken van de marktstandaarden voor testen en testmanagement.

Dat gezegd hebbende zijn we er echter nog niet. Naarmate de informatiesystemen nog complexer worden en nieuwe systeemontwikkelmethoden zich ontwikkelen, zal ook het testdomein mee moeten evolueren. De opkomst van nieuwe trends zoals big data, blockchain en zelfs het genereren van programma's door middel van AI zoals ChatGPT, vraagt om verdere evolutie van het testen. Hopelijk wordt over vijftig jaar in een retrospectief op dit artikel een vergelijkbare conclusie getrokken, en is het testen verder geprofessionaliseerd.

Literatuur

[Urba74] Urbanus, J.H. (1974). De organisatie van het testen in relatie tot de gebruiker. *Compact*, 1974(1), 3-9.

Over de auteur

Dennis Stam MSc CISA EMITA werkt als director bij KPMG Advisory N.V. Hij is verantwoordelijk voor het Tech Advisory-team, dat zich richt op het verbeteren van softwarekwaliteit in brede zin. Hij specialiseert zich daarbij voornamelijk op de onderwerpen broncodeonderzoek, testing en testmanagement, maar heeft ook veel ervaring met andere aspecten binnen systeemontwikkelingstrajecten.