

Inhoud

1	Historie en context	1
	Historie	1
	Context	3
	Waar staat testen in relatie tot nieuwe ontwikkelingen?	5
	Toolboxitems	7
2	Visie op testmanagement	9
	De zachte kanten van testmanagement	10
	Traditioneel of agile testen?	11
	Naamgeving: testsoorten of testvormen?	13
	Testers, de nieuwe universal soldiers?	15
	De toekomst?	16
	Toolboxitems	17
3	Testmanagement, een inleiding	19
	Doel van testmanagement	19
	De vergeten succesfactoren van testmanagement	20
	Waar moet ik allemaal rekening mee houden?	20
	Wat bedoelen ze eigenlijk met....?	22
	Ga er maar aan staan als testmanagement!	34
	Toolboxitems	35
	Fase Voorbereiding	36
4	Een testplan op één A4'tje!	41
	Ervaringen met testplannen	41
	Waarom een testplan?	43
	Een testplan op één A4'tje!	44
	Traditioneel testplan	49
	Agile testplan	52
	En hoe gaat dat in de praktijk?	55
	Samenvatting	58
	Toolboxitems	59
5	Stakeholders	61
	Wat is of zijn stakeholders?	61
	Fijn dat ik dat weet, en wat moet ik dan met die stakeholders?	64
	En hoe pas ik dat toe in het testproces?	67
	Stappenplan stakeholderanalyse testproject	67
	En hoe gaat dat nu in de praktijk?	79
	Samenvatting	82
	Toolboxitems	83

6	Betrokkenheid en draagvlak geven vertrouwen	85
	Betrokkenheid en draagvlak	86
	Hoe creëer je draagvlak?	88
	De B&D-cirkel	88
	En wat betekent dat voor testen?	91
	En hoe gaat dat in de praktijk?	95
	Samenvatting	96
	Toolboxitems	97
7	Een communicatieplan in zeven stappen	99
	Wat is effectieve communicatie voor testmanagement?	99
	Een communicatieplan in zeven stappen	101
	En hoe gaat dat in de praktijk?	108
	Samenvatting	110
	Toolboxitems	111
8	Zonder productrisicoanalyse test je blind	113
	Wat zijn productrisico's en waar bestaan ze uit?	114
	Hoe verloopt het productrisicoanalyseproces?	115
	Specifieke productrisicoanalyse-aspecten voor traditionele omgeving	120
	Specifieke productrisicoanalyse-aspecten voor agile omgeving	121
	En hoe gaat dat in de praktijk?	124
	Samenvatting	128
	Toolboxitems	129
9	Teststrategie bepalen vanuit kennisniveau	131
	Teststrategie algemeen	132
	Wat is een goede teststrategie?	134
	Specifieke teststrategieaspecten voor traditioneel	142
	Specifieke teststrategieaspecten voor agile scrum	148
	Specifieke teststrategieaspecten voor DevOps	151
	En hoe gaat dat in de praktijk?	154
	Samenvatting	157
	Toolboxitems	157
10	Begroten is een invuloefening geworden	159
	Waarom weinig tijd besteden aan begroten?	159
	Hoeveel uur heb je ongeveer nodig?	162
	Hoe zet ik uren om in geld?	163
	Specifieke aspecten voor traditioneel	164
	Specifieke aspecten voor agile scrum	165
	Specifieke aspecten voor DevOps	166
	En hoe gaat dat in de praktijk?	168
	Samenvatting	170
	Toolboxitems	170

11	Planning is een ‘eitje’	171
	Planningsperikelen waar je last van kunt hebben	171
	Specifieke planningsaspecten voor traditioneel	174
	Specifieke planningsaspecten voor agile scrum en DevOps	181
	En hoe gaat dat in de praktijk?	187
	Samenvatting	189
	Toolboxitems	189
12	Dé testorganisatie, een vervagende structuur	191
	Van ‘nerd’ naar ‘Brother in Arms’	192
	Dé testorganisatie, bestaat die dan nog?	203
	Organiseren van testen in een traditionele omgeving	207
	Organiseren van testen in een agile omgeving	211
	En hoe gaat dat in de praktijk?	213
	Samenvatting	225
	Toolboxitems	226
13	Van teststrategie naar testaanpak	227
	Een traditionele testaanpak	227
	Een agile testaanpak	233
	Specifieke Full DevOps-aspecten	245
	En hoe gaat dat in de praktijk?	257
	Samenvatting	272
	Toolboxitems	274
14	Testautomatisering is niet zo eenvoudig	275
	Testautomatisering en testmanagement	276
	Testautomatisering, daar gaat het helemaal niet om!	279
	De piramide voor testautomatisering	280
	Wat kun je niet automatiseren?	286
	Technical debt en (test)automatisering	289
	Toolondersteuning	294
	En hoe gaat dat in de praktijk?	297
	Samenvatting	302
	Toolboxitems	303
	Fase Uitvoering	304
15	Verslag uitbrengen in plaats van rapporteren	305
	Waarom verslag uitbrengen in plaats van rapporteren?	306
	Verslag uitbrengen algemeen	306
	Traditionele omgeving	309
	Agile scrumomgeving	314
	En hoe gaat dat in de praktijk?	320
	Samenvatting	322
	Toolboxitems	323

16	De waarde van bevindingen	325
	Twee geleerde lessen	325
	Misverstanden	326
	Doel bevindingenbeheer	327
	Wat is een bevinding?	328
	Wie zijn stakeholders bij bevindingenbeheer?	329
	Bevindingenadministratie	331
	Een waarzegger worden, testmanagement kan het echt!	337
	Specifieke aspecten traditioneel	340
	Specifieke aspecten voor een agile omgeving	342
	En hoe gaat dat in de praktijk?	344
	Samenvatting	352
	Toolboxitems	353
	Fase Afronding	354
17	Goed overdragen is een project doen slagen	355
	Aan wie draag ik over?	355
	Wat draag ik over?	357
	Specifieke overdrachtsaspecten voor traditioneel	362
	Specifieke overdrachtsaspecten voor agile scrum	362
	En hoe gaat dat in de praktijk?	363
	Samenvatting	365
	Toolboxitems	365
18	Evaluatie is leren en vieren	367
	Geleerde lessen vanuit testmanagement	368
	Specifieke evaluatieaspecten voor traditioneel	369
	Specifieke evaluatieaspecten voor agile scrum	370
	Specifieke evaluatieaspecten voor DevOps	377
	En hoe gaat dat in de praktijk?	377
	Samenvatting	380
	Toolboxitems	380
A	Bronnen	381
	Boeken en artikelen	381
	Sites	382
B	Verklarende woordenlijst	385
C	Agile Manifesto	399
	Dankwoord	402
	Over de auteur	404

Voorwoord

Het volgende zou zo maar eens op jou van toepassing kunnen zijn. Word jij als gevolg van nieuwe werkwijzen als agile, DevOps of andere veranderingen binnen jouw organisatie als projectleider geconfronteerd met het feit dat testmanagement plotseling onderdeel is geworden van je functie of rol? En heb je die ‘magische testwereld’ als projectleider tot nu toe altijd toevertrouwd aan testspecialisten?

Ben jij al jaren werkzaam binnen het traditionele vakgebied testen als tester en krijg je er plotseling testmanagementactiviteiten bij waarin je nog geen ervaring hebt opgebouwd? Word je bijvoorbeeld als testspecialist in een scrumteam ook verantwoordelijk voor het automatiseren van de testen of het gehele testmanagement binnen het project, of mag je het testen in de DevOps-wereld gaan automatiseren?

Weet je niet goed hoe je dat moet aanpakken en vormgeven? Dan ben je echt niet de enige. Zo maar een paar vragen uit de praktijk die actueel zijn op het gebied van een veranderende kijk op testen die zich de laatste tien jaar aan het ontwikkelen is:

- ‘Waar haal ik de kennis vandaan en hoe pas ik die toe?’
- ‘Hoe kan ik testmanagementactiviteiten nu opeens wel uitvoeren als daar tot nu toe door de wol geverfde testmanagers voor zijn ingehuurd?’
- ‘Hoe zit dat met conflicterende belangen tussen testmanagement en mijn projectmanagementrol?’
- ‘Ben ik als testspecialist/developer in het scrumteam nu ineens verantwoordelijk voor alles rondom testen? Daar heb ik helemaal de vaardigheden niet voor. Ik ben geen testmanager maar testspecialist.’

Dit boek gaat je helpen met het pragmatisch invullen van testmanagement met behoud van kwaliteit en maximale risicoreductie. Uitgangspunten daarbij zijn: ‘Goed is goed genoeg’, ‘geen risico – geen test’ en ‘geen tijd verspillen aan sub-optimalisatie!’ Het is geschreven vanuit de wil om testmanagement eenvoudiger te maken voor ‘oude’ testers in een veranderende wereld en niet-‘native’ testers.

Succes met Testmanagement is gebaseerd op jarenlange ervaring als test-, project- en businessmanager opgedaan bij diverse (internationale) organisaties en in verschillende sectoren (overheid, NAVO, financieel, dienstverlening en zorg). *Succes met Testmanagement* is toepasbaar in zowel traditionele als agile projecten en in beheersituaties. Iedereen kan zich testmanagement eigen maken en het wordt niet ingewikkelder dan nodig is. Je hoeft ook niet alles zelf te weten of te kunnen. Samen ben je sterk!

In dit boek volg ik het natuurlijk verloop van verandertrajecten. Het maakt niet uit of het veranderingen zijn die worden doorgevoerd vanuit traditionele projecten of door agile teams. Het zijn allemaal planmatige veranderingen en er zijn altijd drie fasen te herkennen: Voorbereiding, Uitvoering en Afronding. Deze drie fasen vormen de kern van het boek. Wat doe je op het gebied van testmanagement in een fase en hoe pak je dat zo simpel mogelijk aan? Ik geef voorbeelden en tips uit de praktijk die je daarbij helpen. Ik geef expliciet aan dat testmanagementactiviteiten niet meer worden gekoppeld aan een bepaalde rol of functie. Iedereen zou na het lezen van dit boek, in combinatie met de Toolbox, in staat moeten zijn testmanagementactiviteiten vorm te geven en succesvol uit te voeren. Als elke lezer een aantal keren tijdens het lezen denkt: ‘Hé, hier kan ik wat mee’, dan heb ik mijn doel bereikt door een stukje kennis over te dragen of een denkproces in gang te hebben gezet.

Leeswijzer

Dit is een boek voor iedereen die in traditionele en agile omgevingen te maken heeft of krijgt met testmanagementactiviteiten: testmanagers en testers, scrumteams en scrummasters, IT- en lijnmanagement, programma- en projectmanagers, productowners, developers, ontwerpers, (eind)gebruikers en IT-beheerders.

In dit boek wordt in principe niet ingegaan op testtechnieken, testspecificaties, taken en rollen binnen het testvak, daarover is genoeg literatuur geschreven. Uitgangspunt is dat de lezer enige kennis heeft van projectmatig werken en van ontwikkelingen als agile scrum en DevOps.

Het boek geeft een theoretische ondersteuning om testmanagementactiviteiten uit te kunnen voeren in de praktijk, maar het is ook een praktijkboek. Het bevat meer dan honderd direct toepasbare tips uit de praktijk en er wordt aangegeven hoe organisaties op dit moment invulling geven aan testmanagement. Bij het boek hoort een Toolbox waarin per hoofdstuk, en in kleur, naast alle figuren die in het boek staan, templates, voorbeelden en nog meer figuren worden gegeven voor direct gebruik. Ik beveel aan om een hoofdstuk uit het boek te lezen in combinatie met het bijbehorende hoofdstuk uit de Toolbox. Na registratie van dit boek ontvang je een bevestigingsmail met daarin de downloadlink voor de Toolbox.

Het boek geeft geen oplossingen in termen als ‘zo moet het’, maar biedt mogelijkheden om op een bepaalde manier naar testmanagement te kijken in termen van ‘zo zou het ook kunnen’, of geeft een denkrichting aan. Elke organisatie is anders, dus ook testmanagementactiviteiten krijgen binnen een organisatie een ‘eigen kleur’.

Als ik het heb over ‘medewerkers’, dan bedoel ik daar alle mensen in een organisatie mee en niet alleen het operationeel uitvoerende niveau. Dus zowel de medewerkers als het lijn- en projectmanagement, de directie en het bestuur zijn voor mij medewerkers in een organisatie.

Ik heb ervoor gekozen om een groot aantal Engelse termen en afkortingen niet om te vormen naar het Nederlands. Het gaat om algemeen ingeburgerde woorden als stakeholder, business case, going concern, build, operations, development, user, story, design enzovoort. Begrippen die niet voor iedereen gesneden koek zijn worden in bijlage B toegelicht. Deze begrippen zijn in de tekst *cursief* aangegeven als het woord voor de eerste keer wordt gebruikt.

Er zijn vele methoden om systemen te ontwikkelen en te testen. Om de duidelijkheid te bevorderen en de complexiteit te reduceren gebruik ik aan de ene kant van het spectrum het begrip ‘traditionele’ methode voor development volgens de watervalmethode met daaraan gekoppeld het testen volgens het V-model (zie onder andere TMap Next, 2006), en aan de andere kant het *agile* gedachtegoed met de ‘oervorm’ van scrum

(Schwaber & Sutherland, 1995) als meest voorkomende vorm. Daarnaast is vanuit agile ook *DevOps* een belangrijke factor geworden. *DevOps* is geen projectmanagementmethode maar een werkvorm waarbij geïntegreerde teams vanuit *Development* en *Operations* continu verbeterde software leveren vanuit de lijnorganisatie.

Hoewel ik zorgvuldig te werk ben gegaan bij het samenstellen van de Toolboxitems, kan ik niet garanderen dat er geen foutjes zitten in een van de spreadsheets. Ik kan dan ook geen verantwoordelijkheid nemen voor eventuele fouten die optreden bij gebruik van de spreadsheets. Ik heb ze uiteraard wel getest.

In de meeste hoofdstukken wordt in de paragraaf *En hoe gaat dat in de praktijk?* aangegeven hoe organisaties invulling geven aan de theoretische modellen. Ik wil benadrukken dat er in deze beschrijvingen geen sprake is van ‘goed’ of ‘fout’. Het gaat alleen om het weergeven van een manier die organisaties hebben gekozen om in hun specifieke context invulling te geven aan traditionele of agile werkwijzen in combinatie met testmanagementactiviteiten.

Bij het gebruik van foto's heb ik mijn uiterste best gedaan alle personen die erop voorkomen te achterhalen en toestemming te vragen om de foto in het boek te gebruiken. In slechts een tweetal gevallen is dat helaas niet gelukt.

Historie en context

“Testing is an infinite process of comparing the invisible to the ambiguous in order to avoid the unthinkable happening to the anonymous.” – James Bach

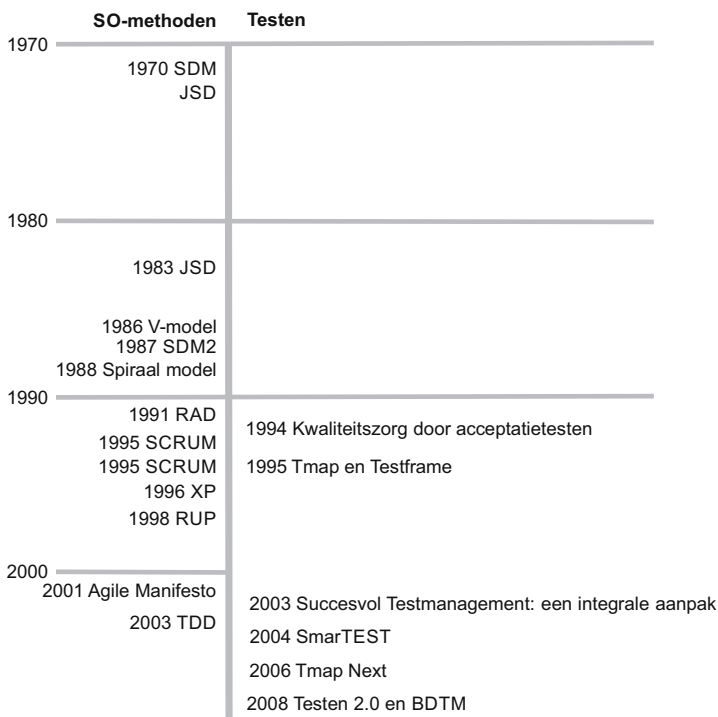
In dit hoofdstuk geef ik een korte schets van de historie en de context waarin testen zich heeft ontwikkeld vanaf ongeveer eind jaren zeventig. Ik kijk kort terug in de geschiedenis van testen in samenhang met developmentmethoden. Dat geeft een beetje gevoel bij waar we vandaan komen. Maar de context is interessant want veranderingen zijn natuurlijk niet voor niets ontstaan. Wat heeft de geschiedenis ons geleerd en gaat het nu – met al die opgedane ervaring in projectmanagement met nieuwe developmentmethoden en gestructureerd testen – ook kwalitatief beter?

Historie

Development en testen zijn door de tijd heen samen opgetrokken in wat je een ‘haat-liefde’-verhouding zou kunnen noemen. Was het in het verleden meer een ‘haat-en-vechtrelatie’, toch heeft de ‘liefde’ uiteindelijk overwonnen. Mede onder invloed van ‘agile denken’ is men naar elkaar toe gegroeid. In de tijdlijn in afbeelding 1.1 staat, niet-limitatief, een aantal belangrijke momenten aangegeven in de geschiedenis van development en testen.

In de jaren zeventig en tachtig is testen letterlijk volgend ten opzichte van development en zeker geen apart benoemde planmatige activiteit. Testen als vakgebied bestaat eigenlijk niet. Nadat het systeem volledig is gebouwd, wordt er ‘getest’ door meestal dezelfde personen die het systeem hebben ontwikkeld. De developer is tevens de tester. De algemene houding in die jaren is dat de specificaties perfect zijn uitgewerkt door gebruik van standaarden en de programmeurs zelf wel kunnen testen als ze klaar zijn met bouwen. Als je geluk had, dan mocht ook een aantal gebruikers nog even naar het systeem kijken en kon daarna de boel naar productie. Er was geen sprake van een gestructureerde testaanpak, geen

SUCCES MET TESTMANAGEMENT



Afbeelding 1.1: Tijlijn developmentmethoden en testen.

testdocumentatie en dus ook geen inzicht in wat er wel en niet werd getest. Een ander gevolg van deze werkwijze is dat bevindingen moeilijk te analyseren of te plaatsen zijn.

In de jaren negentig wordt testen een stuk volwassen. Er wordt gebruikgemaakt van documentatie (het functionele en technische ontwerp) als basis voor de uit te voeren functionele testen en er wordt testdocumentatie opgesteld. Er wordt nog niet echt een belang aangegeven voor de te testen functionaliteit, dus het is onduidelijk of je aan het einde van de testtijd de belangrijkste zaken hebt getest. Verder wordt in die jaren een van de belangrijkste mijlpalen in het gestructureerd testen bereikt. In Nederland worden kort na elkaar TMap (gericht op gestructureerd testen) en Testframe (gericht op geautomatiseerd testen) als methoden geïntroduceerd. Een andere grote 'driver' die testen echt als vakgebied op de kaart heeft gezet, is de datumovergang rond de eeuw-

wisseling (Y2K-problematiek). De – deels ten onrechte – paniek rond de datumovergang naar deze eeuw heeft veel bijgedragen aan de erkenning van het vakgebied testen.

In deze eeuw ontstaan er nieuwe methoden/aanpakken zoals SmarTEST (W-model) en Testen 2.0 (agile testen), die aansluiten bij de nieuwe denkwijze over testen en het *agile gedachtegoed*. Naast de ontwerpdocumentatie wordt er ook steeds meer gewerkt met requirements vanuit de business als basis voor het testen en er wordt gekeken naar product-risico's. Daarmee komt het belang van de business meer naar voren en kan de relatie tussen eisen, aantal bevindingen en acceptatie worden gemaakt. Professionele testers zien we over de volle breedte van development aan het werk, maar ook in specifieke aandachtsgebieden als bijvoorbeeld performance-, load-, stress-, security-, usability-, cloud- en ketentesten.

Context

Toch, alle goede ontwikkelingen ten spijt, blijkt de kwaliteit van de informatiesystemen – vooral de geautomatiseerde delen daarbinnen – door de jaren heen in de praktijk behoorlijk tegen te vallen. Het gevolg: veel mislukte projecten, forse herstelkosten en het niet halen van deadlines of volledig mislukte trajecten. Dus kennelijk zijn we er niet met alleen goede en verbeterde development- en testmethodieken.

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011 – 2015 withIn the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Afbeelding 1.2: Succesgraad overall (The Standish Group, 2015).

Het overzicht van The Standish Group (afbeelding 1.2) geeft een beeld van hoe succesvol traditionele softwareontwikkelingsprojecten zijn. Bron van de tabellen is het Standish Group 2015 Chaos Report. The Standish Group onderzoekt sinds 1994 jaarlijks projecten op de mate van succes, in 2015 zijn wereldwijd 50.000 projecten onderzocht. Succes wordt gemeten aan de indicatoren: binnen de tijd, geld en conform specificaties. De tabellen in afbeelding 1.3 spreken voor zich. Ongeveer 30% van de projecten is succesvol en er lijkt weinig progressie te worden gemaakt. Kennelijk leren wij niet voldoende van mislukte projecten, want deze percentages waren in het verleden vergelijkbaar.

Dus wat is er dan nog meer aan de hand, zit het dan misschien niet in de methode of aanpak maar in het veranderen zelf? Je zou zeggen dat organisaties nu wel gewend zijn aan veranderen. Zou het probleem dan in de IT zelf zitten? Immers, grote veranderingen binnen organisaties zijn bijna niet meer denkbaar zonder dat IT daar een grote rol in speelt. Sterker nog, vaak zijn de veranderingen geïnitieerd onder invloed van het beschikbaar komen van nieuwe IT-mogelijkheden. Organisatieveranderingen doorvoeren doen we al honderden jaren. Toch zie je veel trajecten, waarbij IT als grootste ‘driver’ fungeert, vaak mislukken of slechts voor een deel slagen.

Geen wonder dat er de afgelopen dertig jaar hard is gewerkt om softwareontwikkeling wel tot een succes te maken. Dat heeft geleid tot een scala aan developmentmethoden die elk veel beloven en verbetering van succesvolle projecten voorspellen. Vanaf begin deze eeuw hebben we echt te maken met het *agile gedachtegoed*, met het *scrumraamwerk* en *DevOps* als nieuwe aanpak die projecten en systemen succesvoller moeten maken. In de praktijk zien we dat daadwerkelijke implementatie van scrum pas echt de laatste tien jaar van de grond komt en DevOps begint nu langzaam uit te groeien tot een volwaardig ‘Full’ *DevOps*. In tegenstelling tot het verleden lijkt het nu onder die invloed echt de goede kant uit te gaan.

Uit het onderzoek blijkt dat het, nu de agile ontwikkelmethoden een aantal jaren worden gebruikt, mogelijk is om projectresultaten op basis van agile en waterval te vergelijken (zie afbeelding 1.3). Als je naar alle

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

Afbeelding 1.3: Waterval versus Agile (The Standish Group, 2015).

projecten kijkt, dan is er over de gehele lijn te zien dat een aanpak met agile meer succesvolle projecten aflevert en minder totaal mislukte projecten.

Echter, bij deze cijfers zijn wel een paar kanttekeningen te plaatsen:

- De cijfers zien er minder florissant uit als het team niet goed is opgeleid binnen de agile aanpak (in methode en gedrag).
- En als slechts delen van een organisatie agile werken (bijvoorbeeld alleen de IT) en de overige delen niet, dan heeft dit ook een negatieve invloed op de agile percentages en blijft het net zo moeilijk om succesvolle projecten op te leveren als met de traditionele projectaanpak.

Waar staat testen in relatie tot nieuwe ontwikkelingen?

Testen is de afgelopen 30 jaar meegegroeid met alle vernieuwingen die zijn doorgevoerd in het ontwikkelen van informatiesystemen en heeft zich tot een volwassen vakgebied ontwikkeld. Testen is van een specia-

lisme waar niet altijd iedereen alles van begreep, begrijpt of zou willen begrijpen, een vast onderdeel geworden van het IT-veranderingsproces in organisaties. Zo volwassen dat er binnen de testgemeenschap zelf met een scheef oog wordt gekeken naar nieuwe ontwikkelingen binnen het vakgebied. Regelmatig wordt de vraag gesteld of we weer teruggaan in de tijd en de developers, onder invloed van scrum, weer zelf gaan testen. Wel, sterker nog, er zijn formeel geen echte testers meer in een scrumteam maar 'slechts' developers, die overigens wel een specifiek aandachtsgebied, zoals testen, mogen hebben.

Het is zonder meer waar dat agile grote invloed heeft op de manier waarop het testen wordt vormgegeven. De positieve wending in het realiseren van meer succesvolle projecten zal doorwerken in de manier waarop projecten worden aangepakt. Agile, met als bekendste verschijningsvormen scrum en DevOps, wint steeds meer aan populariteit en raakt testen op vele vlakken. Een grote verandering zal binnen testmanagement plaatsvinden. De rol van traditionele testmanager, het overbekende schaap met de vijf poten, zal binnen agile projecten verdwijnen en testmanagement zal een andere invulling krijgen. Denk hierbij aan een projectleider die tegelijkertijd verantwoordelijk wordt voor testmanagementactiviteiten in een project, en wat dacht je van de testspecialist in het scrumteam of de change en release manager die de testmanagementactiviteiten 'erbij' mogen doen? Is dit vloeien in de testkerk? Dat zou je denken maar het is niet eens toekomstmuziek. Het is al werkelijkheid, ik doe er zelf aan mee, en het bevalt eigenlijk wel. En wat dacht je van testen in productie, ook dat gebeurt al met volle medewerking van iedereen.

Ik zie nu heel veel mensen in de testgemeenschap bedenkelijk kijken. Is die testmanager, die altijd nodig is geweest om programma's, projecten, releases en changes kwalitatief goed naar productie te krijgen ineens overbodig? Lopen die testprocessen en testactiviteiten plotseling vanzelf zonder testmanagement? Maar ook de niet-testers weten niet altijd wat ze met hun nieuwe testmanagementactiviteiten (en dus verantwoordelijkheden) aan moeten. Terechte vragen, en een grote verantwoordelijkheid voor testen om hier richting aan te geven.

Kenmerk van een volwassen vakgebied is dat je verder kijkt dan alleen de ‘incrowd (test)gemeenschap’. Ga al die waardevolle kennis delen met jouw omgeving. Wees een ambassadeur en help de niet ‘native’ testers op het paard. Daar maak je vrienden mee en wordt de toegevoegde waarde van testen vergroot. Deel en maak wijzer!

Testen is meegegroeid met de vernieuwingen en heeft een testaanpak voor elke variant. Daarbij is een apart ingericht testteam of testproject de standaard bij traditioneel ingerichte projecten. Het testen is binnen agile nog steeds in ontwikkeling en zoekt zijn weg om zo optimaal mogelijk de nieuwe methoden en werkwijze te ondersteunen. Dat is ook terug te zien in actuele onderwerpen als scaling agile met behulp van *SAFe*, *Nexus*, *LeSS*, *Scrum of Scrums* of het *Spotify Model*. Daarin wordt gezocht naar optimalisatie voor het werken met meerdere scrumteams die samenwerken aan hetzelfde doel of over de hele breedte op organisatieniveau. Daarnaast is een ontwikkeling zichtbaar waarbij ook *lean* elementen worden toegepast, zoals Kanban binnen DevOps. In diverse hoofdstukken zal aandacht worden besteed aan nieuwe ontwikkelingen en hoe testmanagement daarop in kan spelen.

Testen heeft de starheid van het ‘tot op de komma en punt’ volgen van teststandaarden en -methoden losgelaten. Dit heeft door voortschrijdend inzicht geleid tot flexibeler en pragmatische inzichten als *exploratief testen* en ‘goed is goed genoeg’, waarbij gebruikgemaakt wordt van ervaring en kennis van gebruikers en businessbesluitvorming leidend is. Het geautomatiseerd testen is in veel gevallen nog gericht op de unit- en regressie-testen maar onder invloed van agile en DevOps wordt nu de slag gemaakt om zo vroeg mogelijk geautomatiseerd te testen en feedback te verwerken. Een prima ontwikkeling, want hoe eerder de fouten worden gevonden, hoe goedkoper het herstel. ‘Shift left’ dus, en is dat niet wat wij als testgemeenschap altijd al zo graag wilden bereiken?

Toolboxitems

- Tijdslijn developmentmethoden en testen
- Succesgraad overall
- Waterval versus Agile

Visie op testmanagement

“Laten we het vooral niet moeilijker laten lijken dan het is, en alleen dat doen wat echt nodig is.” – Meindert Munnik, 2018

Als ik kijk naar het eerste deel van de bovenstaande zin, dan zeg ik dat omdat er in mijn optiek te ‘spannend’ wordt gedaan over het fenomeen testmanagement en het zogeheten schaap met de vijf poten, de testmanager uit de traditionele testwereld. Wat is er nou eigenlijk zo speciaal aan testmanagementactiviteiten? Wat heb je nou eigenlijk echt nodig? Zou iedereen die een beetje affiniteit met kwaliteit en projectmanagement heeft die testmanagementactiviteiten niet kunnen uitvoeren? Zou in een scrumteam iemand met een testachtergrond niet die testmanagementactiviteiten, eventueel met wat extra ondersteuning van de scrummaster of productowner, erbij kunnen nemen? Is de bron van werkelijke testkennis juist niet testmanagement maar het specialisme van de mensen die daadwerkelijk met testtechnieken aan de slag zijn, zich in de scrumteams al pokerend door de requirements worstelen of de geautomatiseerde testen maken voor de DevOps-testwereld?

Het gaat bij testmanagement om een beperkt aantal specifieke activiteiten die ergens in de tijd door iemand moeten worden uitgevoerd. Ik ben van mening dat een projectleider met een hart voor kwaliteit (en welke projectleider heeft dat niet?) met het doornemen van een beperkt aantal hoofdstukken uit dit boek, ondersteund door templates, een heel eind zal komen. Een groot deel van de testmanagementactiviteiten zijn activiteiten die ook in projecten of lijnorganisaties worden uitgevoerd: begroten, een plan en planning maken, risico's in kaart brengen, een vorm van verslag uitbrengen, medewerkers selecteren en al of niet extern inhuren. Maar ook de testprofessional die naast zijn vroegere testactiviteiten nu ook testmanagementactiviteiten erbij ‘mag’ gaan doen als ‘lead tester’, kan leren juist die andere elementen onder de knie te krijgen.

Als ik het tweede deel van de eerste zin neem, zie je gelijk mijn warme gevoel voor het toepassen van gezond verstand op testen. Het testproces

is uitermate geschikt om continu te verbeteren en verspilling te ontdekken en weg te werken. Traditioneel is procesverbetering gericht op het optimaliseren van de activiteiten, maar binnen *lean* is de procesverbetering gericht op het opruimen van activiteiten die geen waarde toevoegen. Oké, er zijn ook activiteiten die geen waarde toevoegen, maar wel belangrijk zijn omdat je moet voldoen aan wetgeving of specifieke standaarden. Die activiteiten voeg je dan vanuit die verplichting toe.

Activiteiten die geen waarde toevoegen vanuit testmanagement, zijn die er dan? Wat dacht je van het dubbel uitvoeren van testen (systeemtest en functionele *acceptatietest*) omdat er nu eenmaal een andere verantwoordelijkheid bij hoort in de traditionele omgeving (leverancier versus opdrachtgever)? Dat kan best verbeteren door dat anders aan te pakken. Of van bergen testdocumenten die meer gericht lijken te zijn op 'indekken' dan op efficiënte vastlegging van wat nodig is? Denk hierbij aan dikke testplannen voor releases die elke keer hetzelfde zijn – een hoog 'copy-paste'-gehalte hebben – en waar alleen de inhoud van de release in het plan wordt aangepast en op basis daarvan de teststrategie. Dat kan best simpeler. Een testplan kan echt op één A4'tje.

Kortom, testmanagement richt zich op adviseren over kwaliteit, en als het even kan de voorspelbaarheid daarvan, en de manier waarop dat zo effectief mogelijk kan worden bereikt.

De zachte kanten van testmanagement

Ik ben verder van mening dat ingezette verbeteringen nog veel betere resultaten zouden kunnen opleveren als 'zachtere kanten' van projectontwikkeling en testen meer aandacht zouden krijgen. Veel mislukte projecten worden veroorzaakt door een slechte communicatie met de stakeholders (als het al gelukt is die allemaal in beeld te krijgen). Daardoor voelen mensen zich miskend, vergeten en niet serieus genomen. Het gevolg daarvan is dat er geen of onvoldoende betrokkenheid en dus onvoldoende draagvlak voor testmanagement is of blijft. Het voorgaande in combinatie met het veel te grote optimisme dat de mens eigen is als het gaat om het inschatten in welke tijdspanne iets kan worden gerealiseerd, blijft ervoor zorgen dat projecten niet zo succesvol zijn als zou

kunnen. Dit is ook de reden dat ik relatief veel aandacht besteed aan het realiseren van betrokkenheid en draagvlak bij stakeholders, onderwerpen die binnen testmanagement meestal zeer beperkt aan de orde komen. En het is natuurlijk ook niet voor niets dat het belang van communicatie van het Agile Manifesto (bijlage C) afdruipt.

Traditioneel of agile testen?

Sorry, geen voorkeur! Het gaat namelijk helemaal niet om die keuze, het gaat erom wat past, wat het beste aansluit in een bepaalde situatie. Wat mij betreft is er geen enkel probleem om binnen grotere organisaties traditioneel en agile testen naast elkaar te laten bestaan. Ook al lijkt de één star en methodevolgend te zijn en de andere agile en autonoom. Wees je overigens wel bewust dat bij beide vormen op voorhand het tijdspad, budget en de requirements (al is het op hoog niveau) vooraf worden vastgelegd.

Natuurlijk is het goed dat er nu eindelijk werk wordt gemaakt van scrum (dat al dateert van midden jaren negentig) en DevOps. En natuurlijk is het goed dat er vanuit testmanagement wordt nagedacht over hoe je daar het beste op kunt aansluiten qua testactiviteiten. Maar al het goede ten spijt, ook deze ontwikkelingen worden weer commercieel gebracht als ‘de ultieme oplossing’, de heilige graal voor alle problemen. De nieuwe agile aanpak zou ook ‘beter’ zijn dan de traditionele watervalaanpak van projecten. Vergeet het maar, ook deze vernieuwingen zullen bijdragen aan verbetering en kwaliteit van bepaalde soorten informatiesystemen, maar echt niet alle problemen oplossen. En echt, niet alles is op te lossen met behulp van een vrij rigide systeem van scrumsprints waarbij alleen de functionaliteit flexibel is. Als je de in sprintbacklog geplande voorgenomen *stories* niet realiseert, en ze komen op de een of andere manier later terug in een andere sprint, dan noem ik dat gewoon uitloop.

De traditionele manier van development wordt nu bijna in de ban gedaan en als organisatie ben je echt ‘old school’ als je nog traditionele methoden gebruikt voor development. Maar iedereen die in zo’n ‘beschamende’ situatie werkt, steek ik hierbij een hart onder de riem. De traditionele manier van development zal voorlopig niet verdwijnen. Ster-

ker nog, het kan wel eens zo zijn dat dit juist de perfecte methode is in een bepaalde situatie.

Blijf vooral realistisch en nuchter kijken naar waar welke aanpak het meest geschikt is in een bepaalde situatie. Stel jezelf de vraag: wanneer kies ik voor agile en wanneer voor de traditionele watervalbenadering? In mijn visie is het niet de projectmanagementmethode die het succes bepaalt van development maar het maken van een keuze voor het een of het ander. Maak bijvoorbeeld een keuze op basis van de onderstaande argumenten en richt je testmanagementactiviteiten op basis van die keuzes optimaal in.

Kies voor een agile teststrategie als:

- Het moeilijk is om vooraf alle eisen SMART te definiëren.
- De verwachting is dat er veel scopewijzigingen in het project komen.
- Het volledige project niet vooraf is te plannen.
- Er snel resultaten nodig zijn die businesswaarde genereren.
- De organisatie ook echt mensen vrij kan maken voor agile werkende teams.
- Er softwaresystemen worden ontwikkeld of onderhouden die een directe relatie met de klant hebben (online/web).

Kies voor een traditionele watervalteststrategie als:

- Er bij de start van het project een betrouwbare budgettering nodig is.
- Er een groot belang is bij het goed documenteren.
- Het een erg groot project (programma) met een hoog risico is dat bestaat uit een groot aantal deelprojecten met onderlinge afhankelijkheden.
- Tussenmijlpalen gewenst zijn.
- Het een verplichting is (wetgeving, standaarden vanuit de overheid).
- Er door externe partijen softwaresystemen worden geleverd die gericht zijn op bestuurlijke informatieverwerking (backoffice) en via een vast releaseschema worden onderhouden. De releases zijn optie voor een 'hybride scrum'-vorm. Die laatste heb ik niet van mijzelf maar klinkt wel goed.

Er wordt vaak vergeten dat er nog veel organisaties zijn die via de traditionele waterval hun development en testen zullen (moeten) uitvoeren omdat dit het beste past in de omgeving waarin zij opereren. Denk hierbij aan organisaties die veel moeten vastleggen vanuit controle/beheeraspecten zoals overheid en financiële instellingen. Er zijn ook projecten die vanuit risico-oogpunt niet ‘gaandeweg’ tot de betere of beste oplossing kunnen komen zoals in agile. In die omgeving moet juist op voorhand alles goed worden gespecificeerd om grote ongelukken te voorkomen. Internationale organisaties, en daarbinnen de programma’s en projecten, werken vaak verplicht met (internationale) standaarden. Ook als er mensenlevens mee gemoeid kunnen zijn als het fout gaat, wordt er vaak eerst gespecificeerd, dan gebouwd en getest tot in het oneindige (in vergelijking met bestuurlijke informatiesystemen). Een agile ontwikkeld vliegtuig zou moeten kunnen, maar je zult als project toch moeten voldoen aan internationale luchtvaartstandaarden, wat je ook als team bedenkt en hoe blij de productowner ook met het resultaat is. Ook een draagbare automatisch werkende medicijnverstrekker (ten behoeve van een chemokuur of pijnbestrijding) zal eerst heel nauwkeurig worden gespecificeerd, gebouwd en getest. Daar telt het statement *No risk – No test* niet en zal elk requirement of gespecificeerde functionaliteit en onderkend productrisico maximaal worden getest. Terecht natuurlijk, er zijn mensenlevens mee gemoeid.

Naamgeving: testsoorten of testvormen?

In de praktijk is het niet altijd helder over welke testen we het hebben omdat er gebruik wordt gemaakt van begrippen als testsoort en testvorm. Met de opkomst van agile terminologie wordt het er niet duidelijker op. Ik heb mijzelf afgevraagd of al die specifieke benamingen in soorten en vormen wel nodig zijn en ben tot de conclusie gekomen dat misverstanden kunnen worden voorkomen als je kiest voor een eenvoudige benadering. Om verwarring te voorkomen los ik dat op door niet te kiezen voor een onderscheid in testsoorten en -vormen en alleen het begrip testsoort te gebruiken. Dan ben je daar alvast van af.

Ik onderken de volgende drie testsoorten:

- Ontwikkeltesten: testen die de developer al of niet geautomatiseerd op de door hem ontwikkelde code uitvoert om te kijken of die code technisch werkt.
- Systeemtesten:
 - Testen waarin door de leverancier al of niet samen met een agile team of traditioneel testproject onderzocht of het systeem inclusief de in- en externe interfaces werkt.
 - Testen die door een agile team worden uitgevoerd en waar, tijdens de testuitvoering, constant feedback wordt ‘opgehaald’ bij de gebruiker door deze de oplossingen te laten beoordelen.
- Acceptatietesten: testen waarin de acceptant, al of niet vanuit of samen met een agile team, onderzoekt of het systeem voldoet aan de door de business gestelde requirements. Ook het testen in productie is voor mij een acceptatietest. Hoewel daar meestal minder stringent requirements voor zijn aangegeven, gaat het meestal over het beoordelen en monitoren van een aantal kwaliteitskenmerken.

Wanneer is een testsoort geschikt voor een bepaalde test? Dat is afhankelijk van een aantal aspecten. Meestal hangt dat samen met de vormgeving van de verantwoordelijkheden of contractstelling (alles in eigen hand of een verhouding leverancier <> opdrachtgever). Dat bepaalt meestal waar je naar wilt kijken tijdens de uit te voeren testen (test je de code of keten?) en welke kwaliteitskenmerken er uit de productrisicoanalyse naar voren zijn gekomen (test je performance, functionaliteit, middelenbeslag of bruikbaarheid?).

Met deze indeling kun je zowel in de traditionele wereld als binnen agile aan de slag. Het feit dat er een indeling in drie testsoorten is, wil niet zeggen dat alle drie de testsoorten ook in een teststrategie moeten worden gebruikt. Je kunt je bijvoorbeeld afvragen of je nog wel over systeem- en functionele acceptatietesten wilt praten of dat die testsoorten worden gecombineerd tot één uit te voeren acceptatietest die wordt gevolgd door een beperkte acceptatietest gericht op de specifieke organisatie-inrichting. En logisch, er is natuurlijk niet zoiets als dé ontwikkel-, systeem- of acceptatietest. Er zijn talloze aantal testen die in een van deze

drie categorieën vallen of in meerdere testsoorten kunnen worden gebruikt (zie het overzicht in de Toolbox behorende bij dit hoofdstuk).

Testers, de nieuwe universal soldiers?

Misschien nog bekend onder de wat oudere jongeren? De film *Universal Soldier* (1992), waarin in Vietnam gedode soldaten worden bevroren en naar een geheim onderzoekscentrum in Amerika worden gevlogen, om in een laboratorium te worden omgebouwd tot supersoldaten: universal soldiers. Als ik nadenk over testen moet ik vaak aan deze film denken. Het is voor mij een beeld waarbij ik de voorspelde ontwikkelingen van ‘testen’, zoals het zo vaak wordt genoemd, langs zie komen.

De introductie van een nieuwe developmentmethode gaat vaak gepaard met de mededeling dat ‘alles’ vanaf nu anders (lees: veel beter) zal gaan. Vooral ‘het testen’ zou sterk veranderen en testers zouden niet meer nodig zijn (invriezen). Hoe vaak heb ik al niet gehoord dat ‘de developer’ vanuit de nieuwe methode wel zelf gaat testen en echte professionele testers overbodig zouden worden. En als zo’n nieuwe aanpak eenmaal een beetje was ingeburgerd, ging het toch niet helemaal ‘chill’ met de kwaliteit. En dan worden de testers weer opgetrommeld (ontdooien), krijgen een update door middel van een opleiding en gaan weer aan de slag. En zie, de wedergeboorte van de nieuwe versie ‘universal test soldier’ 2.0 is daar. In het huidige agile scrum- en DevOps-tijdperk is ‘testen’ wederom in de situatie beland dat in veel organisaties wordt gedacht dat ‘het testen’ ook wel zonder de universal test soldier kan (en weer invriezen). De projectmanager als testmanager, de businessinformatieanalisten en applicatiebeheerders als testontwerpers en testuitvoerders. Iedereen een testopleiding en klaar zijn we voor de komende decennia. Ik voorspel alvast dat ook de agile wereld niet zo succesvol zal zijn als die zou kunnen zijn zonder ‘universal test soldiers’.

Vanuit de business, agile coaches maar ook uit scrum- en DevOps-teams, waar iedereen ‘alles’ moet kunnen, komen signalen dat de kwaliteit toch wel te wensen over laat. Er wordt ‘te veel gemist’ bij het testen. De vraag wordt gesteld hoe dat kan: ‘We hebben het toch goed vastge-

legd in de Definition of Done, waar gaat het nu mis?' Wel, misschien ontbreekt hier wel de universal test soldier x.0?

Waarom het testen zonder echte professionele testers niet zo goed zal gaan lukken, heeft te maken met het verschil in houding van developers en testers. De developer wil nieuwe dingen maken, innoveren en stelt zich de vraag of het werkt en niet de vraag of het fout kan gaan. En het werkt als het goed is gecodeerd. De professionele tester wil zo veel mogelijk kwaliteit leveren en stelt bij elke oplevering kritische vragen: gaat dit werken, gaat dit altijd zo werken, zou het niet kunnen dat....? Stel aan een scrumteam als er tijd overblijft in een sprint de volgende vraag: zullen we wat extra gaan testen om de kwaliteit te verhogen of zullen we nog een user story oppakken? Ik ken het antwoord en daarom zijn er universal test soldiers nodig.

Er is overigens een vervolg op deze film gemaakt: Universal Soldier II: Brothers in Arms. En dat lijkt mij voor de huidige testers die, met extra uitrusting door opleiding en vorming, in agile scrum- en DevOps-teams werken, een perfecte uitdrukking voor de nieuwe universal test soldiers: Brothers in Arms. Overigens zijn van elke vorige versie van de universal test soldiers exemplaren in de vriezer bewaard. Je weet maar nooit wanneer je een oudere versie weer uit de vrieskast moet halen.

De toekomst?

Quality in agile Teams – Testing in the driver's seat. Dat is de titel van een in oktober 2017 door KPMG gepubliceerd rapport. Het geeft, vrij vertaald, aan dat de kwaliteit van agile teams nauw samenhangt met de wijze waarop invulling wordt gegeven aan testen. Testen vindt steeds meer plaats aan de voorkant van het developmentproces en voegt de meeste businesswaarde toe aan de organisatie. Steeds meer agile teams kiezen voor 'test driven development' (TDD) en een acceptatietest gedreven aanpak. Sinds organisaties agile zijn gaan werken is geautomatiseerd testen, over alle sectoren, toegenomen met vijfentachtig procent. Testautomatisering is een dominante 'must have' om snel te kunnen anticiperen op verandering.

De tester die vandaag de dag waarde wil toevoegen aan agile teams, heeft niet genoeg aan alleen een testtechnische invalshoek, maar zal meer en meer georiënteerd moeten zijn op het vervullen van verwachtingen en het toevoegen van waarde voor de klant. Het vinden van fouten en gebreken alleen is niet meer voldoende. De tester zal ook een testresultaat moeten kunnen interpreteren, of het compleet en acceptabel is voor de klant. Het rapport geeft ook aan dat er daarnaast nog veel sectoren en gebieden zijn waar zeer waarschijnlijk met een meer traditionele aanpak betere resultaten zullen worden geboekt. Dat is niet alleen gerelateerd aan meer functioneel georiënteerd zijn als bedrijf, maar ook de context speelt een belangrijke rol: veranderingsbereidheid, cultuur van een organisatie, is agile een oplossing voor de oorzaak van problemen, de soort afnemer, wetgeving, verantwoording.

Testing in the driver's seat, de droom van elke echte tester.

Wat valt er verder nog te zeggen?

Toolboxitems

- Overzicht met gangbare testen
- Overzicht met agile methoden