

MCTTL

En de houthakker hakte tot diep in de nacht,
want hij had geen tijd om zijn bijl te slijpen.

Hoofdstuk 6.4

Taakgebied Testen

V1.19.1 / 01 februari 2019

Auteur: Ton van den Hoogen

Met dank aan alle bedrijven en personen die in de afgelopen jaren bewust en onbewust een bijdrage aan MCTL hebben geleverd.

Tekstredactie: TekstFontein



Geen copyright!

MCTL is in licentie gegeven volgens een Creative Commons Naamsvermelding 3.0 Nederland licentie. Gebaseerd op een werk van www.mctl.nl.

MCTL is geheel Public Domain, er rusten dus *geen* copyrights of auteursrechten op. U mag MCTL (ook commercieel) gebruiken, verwerken, bewerken ... wat u maar wilt. Wanneer iets echter Public Domain is, blijft het Public Domain. Wat u dus niet mag doen is over (delen van) MCTL copyright of auteursrechten claimen, u maakt zich dan schuldig aan copyfraud en bent strafbaar. Indien u zelf overtredingen constateert, vragen wij u dit via www.mctl.nl aan ons te melden.

Wat wij van u vragen is om bij elk gebruik een verwijzing naar de bron: www.mctl.nl op te nemen. De reden hiervan is dat op deze wijze iedereen de oorspronkelijke versie(s) kan vinden.

MCTL – 6.4. Taakgebied Testen v1.19.1

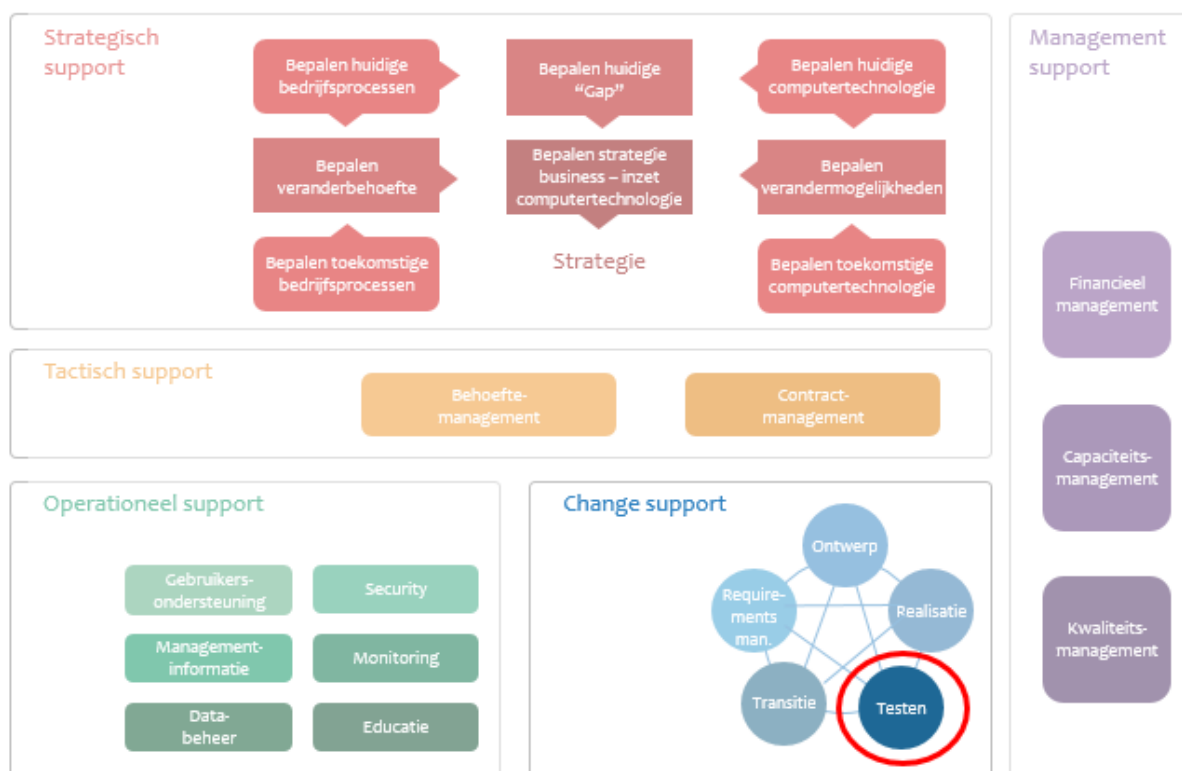
Hoofdstuk 16	Taakgebied Testen	4
	Plaats in het MCTL-framework.....	4
	Achtergrond	4
	Definities.....	5
	Doel van dit taakgebied	6
	Hoe weet je dat het doel is bereikt?.....	6
	Taken 6	
1.	Toetsen specificaties en opgeleverde ontwerp-producten	7
2.	Check op door infra-, applicatiesupport en leveranciers uitgevoerde tests.....	8
3.	Functionele acceptatietest	9
4.	Gebruikersacceptatietest.....	11
5.	Productieacceptatietest.....	14
6.	Opstellen vrijgave-advies/verbeterpunten	16
	Relaties met andere onderdelen van MCTL	16
	Opmerkingen	17
1.	Testsoorten en -vormen	17
2.	Testen met productiedata	19
3.	Testen in productie.....	19
4.	Testen in productie-lookalike-omgevingen	20
	Certificering/proefexamenvragen	21
1.	MCTL Foundation - proefexamenvragen	21
2.	MCTL Foundation – proefexamenvragen met antwoorden en uitleg.....	22
3.	MCTL Advanced-basis - proefexamenvragen	24
	Nuttige websites en boeken	25

HOOFDSTUK 6.4. TAAKGEBIED TESTEN

De werkzaamheden in het vakgebied Testen strekken zich uit van diep in de techniek tot diep in de gebruikersorganisatie. MCTL neemt uitsluitend het functionele deel in ogenschouw. De invalshoek binnen MCTL is niet het gewijzigde systeem, maar de wijzigingen in het bedrijfsproces. Dit businessperspectief betekent dat er getest wordt vanuit een beoogd gewijzigde werkwijze/gewijzigd bedrijfsproces in samenhang met de daarvoor aangepaste computertechnologie: software, databases en/of hardware.

PLAATS IN HET MCTL-FRAMEWORK

Het taakgebied Testen maakt deel uit van het taakcluster Change support.



ACHTERGROND

Testen kan worden gezien als onzekerheidsreductie. Met precies de juiste input en uitvoering van de aanpassingen, zou testen overbodig zijn. Helaas kan vanaf het initiëren van een wijziging tot en met de realisatie ervan van alles misgaan. Via testen is alleen in theorie 100% zekerheid te krijgen dat de gerealiseerde wijziging 100% correct is. Praktisch gesproken is dat niet zo, maar het uitvoeren van testen kan wel aantonen dat bepaalde onderdelen in orde zijn. Hoe beter de testomgeving en -data, het testscript en -team zijn,

hoe groter de zekerheid dat de gerealiseerde wijziging in productie aan de verwachting voldoet.

DEFINITIES

Binnen het taakgebied Testen worden verschillende termen gebruikt. Er wil nog weleens verwarring zijn over welke term waarvoor staat. Daarom worden ze hieronder gedefinieerd (gebaseerd op TMap).

Definitie toetsen

Toetsen is het beoordelen van tussenproducten in het gehele wijzigingstraject.

Definitie testen

Testen omvat activiteiten die inzicht geven in en adviseren over de kwaliteit van eindproducten en de daaraan gerelateerde risico's.

Definitie testsoort

Een groep testactiviteiten die gezamenlijk wordt uitgevoerd en aangestuurd.

Definitie testvorm

Een groep testactiviteiten dat als doel heeft het informatiesysteem op een aantal samenhangende kwaliteitsattributen te controleren.

Definitie functionele acceptatietest (FAT)

De functionele acceptatietest is een door functioneel specialisten uitgevoerde test in een zo goed mogelijk op de productieomgeving lijkende omgeving. Deze test moet aantonen dat de uitgevoerde wijzigingen aan de functionele eisen voldoen.

Definitie gebruikersacceptatietest (GAT)

De gebruikersacceptatietest is een door (een vertegenwoordiging van) de toekomstige gebruikers uitgevoerde test in een zo goed mogelijk op de productieomgeving lijkende omgeving. Deze test moet aantonen dat de uitgevoerde wijzigingen aan de wensen/eisen van de gebruikers voldoen.

Definitie productieacceptatietest (PAT)

De productieacceptatietest is een door (een vertegenwoordiging van) de toekomstige beheerders uitgevoerde test in een zo goed mogelijk op de productieomgeving lijkende omgeving. Deze test moet aantonen dat de uitgevoerde wijzigingen aan de wensen/eisen vanuit beheer voldoen. Beheer kan hierbij zowel de technische als functionele kant betreffen, hoewel de nadruk vaak sterk op de technische kant ligt.

Definitie acceptatiecriteria

Acceptatiecriteria zijn meetbare condities waaraan voldaan moet zijn om een aangepast systeem in productie te nemen. Het opstellen en onderhouden van acceptatiecriteria vindt plaats in Kwaliteitsmanagement. Het taakgebied Testen is een belangrijke 'gebruiker' van de acceptatiecriteria.

MCTL – 6.4. Taakgebied Testen v1.19.1

DOEL VAN DIT TAAKGEBIED

Testen omvat het controleren of de gerealiseerde wijzigingen voldoen aan:

- de gedefinieerde behoefte en specificaties;
- de gestelde kwaliteitsnormen;
- de vanzelfsprekende (dus niet expliciet gedefinieerde) behoeften en specificaties.

HOE WEET JE DAT HET DOEL IS BEREIKT?

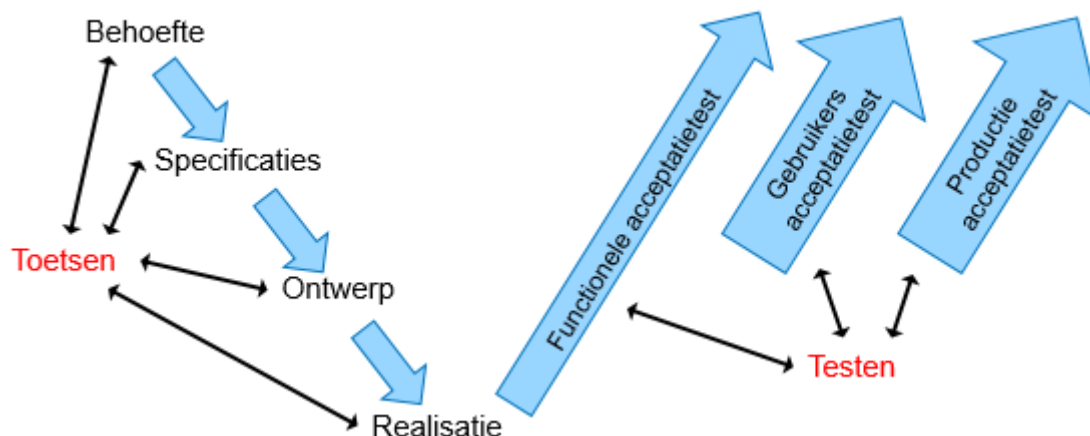
De volgende indicatoren geven aan of bovenstaand doel bereikt is.

- Het aantal fouten dat is geconstateerd nadat de wijzigingen in productie zijn genomen, is < 2% van het aantal bevindingen die in de verschillende tests zijn geconstateerd en verholpen.
- De totaal aan tests bestede tijd/hoeveelheid geld is < 50% van de schade die zou zijn ontstaan wanneer de testbevindingen niet vóór inproductienamen waren hersteld.

(Genoemde percentages zijn indicatief.)

TAKEN

De taken in dit taakgebied zijn als volgt schematisch weer te geven.



Toetsen vindt plaats op de diverse tussenproducten en testen op de diverse eindproducten.

De taken zijn concreet:

1. Toetsen van specificaties en opgeleverde ontwerpproducten
2. Check op door infra support, applicatiesupport en leveranciers uitgevoerde tests
3. Functionele acceptatietest
4. Gebruikersacceptatietest
5. Productieacceptatietest
6. Opstellen vrijgave-advies/verbeterpunten

MCTL – 6.4. Taakgebied Testen v1.19.1

Een toelichting daarop volgt hieronder.

1. TOETSEN VAN SPECIFICATIES EN OPGELEVERDE ONTWERP-PRODUCTEN

Toetsen, het controleren van tussenproducten, kent verschillende technieken. In de context van MCTL zijn de navolgende het meest bruikbaar.

1. Inspecties: hierbij wordt een product beoordeeld dat 100% gereed is, maar nog niet definitief is vastgesteld. Vooraf zoekt een team gestructureerd naar discutabele issues in de te inspecteren documenten. Tijdens een inspectiebijeenkomst worden de bevindingen van het team besproken en dient overeenstemming over deze bevindingen te ontstaan. Men bespreekt nieuwe issues die tijdens de bijeenkomst naar voren gebracht worden. Ook daarover dient overeenstemming te ontstaan. Na de bijeenkomst past de auteur het document aan en kan eventueel een herinspectie plaatsvinden.
2. Reviews: worden gedaan op producten die nog niet helemaal af zijn (60-80% gereed). Een review richt zich op het vinden van fouten en op het verbeteren van het document door de kennis en vaardigheden van degenen die de review uitvoeren. Zij mogen zich dan ook met de oplossingsrichtingen bemoeien en voorstellen voor verbetering doen.
3. Walkthrough: hierbij wordt het document, dat voor 50-100% gereed is, stapsgewijs doorlopen. Daarbij kan de auteur een toelichting en bijvoorbeeld de keuzemogelijkheden en overwegingen daarin geven. De andere deelnemers kunnen direct reageren. Het is een zeer interactieve manier van werken, waarbij wel het gevaar van ad-hoc discussies bestaat.

Aan de hand van bovenstaand schema zijn in de verschillende taakgebieden van het taakcluster Change support de volgende toetsmomenten te definiëren.

Requirements management/specificaties

Het achterhalen van requirements en vervolgens opstellen van specificaties is het onderdeel waarin het bijzonder zinvol is om input te krijgen uit zo veel mogelijk relevante bronnen. Bij Toetsen is het mogelijk via reviews en walkthroughs fouten te achterhalen en inzichten mee te nemen van degenen die deelnemen aan de review/walkthrough. Bovendien kan hier door anderen worden gecheckt of niet te snel naar één bepaalde uitwerking/oplossing is toegewerkt, of consequenties per oplossingsrichting inderdaad allemaal in beeld en afgewogen zijn en of er vanuit infra-, applicatiesupport en eventueel de leverancier voldoende inbreng is geweest.

Ontwerp

In Ontwerp worden een aantal modellen, diagrammen en beschrijvingen bijgewerkt op basis van de in Requirements management opgestelde specificaties. Deze bijgewerkte ontwerpelementen kunnen via inspectie worden getoetst. Wanneer bijvoorbeeld een onderdeel van het systeem nieuw is of sterk wordt gewijzigd, kan het toetsteam via reviews en walkthroughs worden betrokken bij het bedenken van oplossingen. De inhoudelijke controle op juistheid en volledigheid dient echter altijd de overhand te houden.

Realisatie

Hier kan een check worden uitgevoerd of alle (gewijzigde) configuratie-instellingen en parameters zijn beschreven (volledigheid) en correct zijn. Een check op onderlinge strijdigheid en aannemelijkheid is vooral aan te bevelen als bijvoorbeeld parameters of hun onderlinge verhoudingen heel sterk wijzigen. Inspectie is een hier voor de hand liggende techniek.

Transitie

In het taakgebied Transitie wordt een transitieplan opgesteld. Binnen Testen kan het bijgewerkte transitieplan worden getoetst voordat dit binnen Transitie wordt uitgevoerd. Ook hier kan inspectie goede resultaten opleveren.

Eindresultaat is steeds een gestructureerde terugkoppeling naar de opstellers van de specificaties, ontwerpen, plannen etc. Afhankelijk van de omvang en aard van de terugkoppeling kan worden afgesproken dat de aanmerkingen worden verwerkt en nogmaals getoetst of dat een hertoetsing niet nodig is.

2. CHECK OP DOOR INFRA SUPPORT, APPLICATIESUPPORT EN LEVERANCIERS UITGEVOERDE TESTS

Het echte testen start met een check op de door infra support, applicatiesupport en leveranciers uitgevoerde tests. In TMap termen spreken we in dit verband van de unittest (UT), de unitintegrietest (UIT) en de systeemtest (ST). Volgen infra- en applicatiesupport een agile-achtige werkwijze, dan zijn de verschillende tests soms wat minder eenvoudig te onderscheiden. Voorop blijft staan dat iemand die iets creëert (bijvoorbeeld een programmeur), allereerst zijn eigen werk zal moeten controleren. Daarna pas komen degenen in beeld die het resultaat van dit werk zullen gaan gebruiken (gebruikers en functioneel specialisten). Hier speelt wel mee dat infra-, applicatiesupport en leveranciers vaak meer testfaciliteiten ter beschikking hebben dan een gebruikersorganisatie. Voorbeelden van dergelijke faciliteiten zijn tools waarmee data in een testdatabase kunnen worden gemanipuleerd en het simuleren van een grote workload op een systeem.

Van infra-, applicatiesupport en leveranciers mag worden verwacht dat de UT, de UIT en de ST op voldoende niveau worden uitgevoerd. Binnen deze testsoorten kunnen verschillende testvormen worden onderscheiden. Uiteraard moeten de wijzigingen, zowel technisch als tot op zekere hoogte functioneel, door infra-, applicatiesupport en leverancier worden getest. Aan het eind van dit taakgebied vindt u een uitgebreide omschrijving van alle testsoorten en -vormen, zoals de back-up & restoretest, de technische conversietest, de penetratie-/beveiligingstest, de herstartbaarheidstest, de installatietest, de de-installatietest en de uitwijktest.

De scheidslijn tussen waar de verantwoordelijkheden van infra-, applicatiesupport en leverancier ophouden en die van functioneel support/gebruikersorganisatie beginnen, is nog niet zo heel eenvoudig te bepalen. Nu volgen een aantal voorbeelden ter illustratie hiervan.

1. Een softwareleverancier voert regelmatig updates uit op de software die hij levert. Omdat hij niet tot in detail op de hoogte is van de specifieke technische inrichting van alle klantomgevingen, heeft hij een interne referentieomgeving gecreëerd. Deze softwareleverancier kan worden aangesproken op het niet juist (technisch en functioneel) functioneren van de software in die referentieomgeving, maar niet op het incorrect functioneren in een klantomgeving die hier technisch van afwijkt.
2. Een softwareleverancier levert via een cloudoplossing feitelijk geen softwaresysteem meer, maar een werkende functionaliteit. In dit geval is deze leverancier verantwoordelijk voor het technisch en functioneel volledig juist functioneren van het geheel.
3. Een klantorganisatie besluit een standaardsoftwarepakket functioneel geheel afwijkend in te zetten. Infra-, applicatiesupport en leverancier doen aanpassingen op het pakket, maar de klantorganisatie zal zelf moeten controleren of de afwijkende werkwijze na die aanpassingen nog steeds wordt ondersteund door het pakket.

Om duidelijkheid te scheppen in de scheiding tussen werkzaamheden van infra-, applicatiesupport en leverancier enerzijds, en functioneel support en de gebruikersorganisatie anderzijds, kan met een zogenaamde *intake* worden gewerkt. Zodra iets wordt opgeleverd door infra-, applicatiesupport en/of leveranciers, wordt door functioneel support een intake gedaan waarbij een aantal vooraf vastgestelde elementen wordt gecheckt. Op die wijze is al snel duidelijk of de oplevering 'retour afzender' kan worden gestuurd of functioneel support met de functionele acceptatietest aan de slag kan gaan. Het is immers niet de bedoeling dat functioneel specialisten en/of gebruikers testen gaan uitvoeren die feitelijk door infra-, applicatiesupport of leveranciers hadden moeten worden gedaan.

3. FUNCTIONELE ACCEPTATIETEST

In het algemeen wordt de functionele acceptatietest uitgevoerd door functioneel specialisten. De omgeving waarin de test wordt uitgevoerd, lijkt zoveel mogelijk op de uiteindelijke productiesituatie. Er dient een testplan te zijn opgesteld waarin de planning, de te testen onderdelen en een beschrijving van de testscripts en -data zijn opgenomen. Input voor dit testplan komt uit de taakgebieden Requirements management en Ontwerp. Daarnaast dienen de acceptatiecriteria uit het taakgebied Kwaliteitsmanagement ter beschikking te staan.

De functionele acceptatietest moet aantonen dat de uitgevoerde wijzigingen aan de functionele en niet-functionele eisen voldoen. Er wordt dus getest ten opzichte van de gedefinieerde en gespecificeerde eisen in Requirements management en Ontwerp. De neiging bestaat sterk te focussen op de functionele eisen in de verwachting dat infra- en

MCTL – 6.4. Taakgebied Testen v1.19.1

applicatie support de niet-functionele eisen voor hun rekening hebben genomen. Echter, als de we de niet-functionele eisen rubriceren:

- Security
- Continuïteit
- Beschikbaarheid
- Capaciteit

Dan zijn dit zeker ook aspecten die in de navolgende testen naast de functionele aspecten op diverse plaatsen aan de orde kunnen komen. Bijvoorbeeld in een ketentest verdient security een aparte plaats. Maar ook capaciteit kan in een ketentest onderwerp van aandacht zijn om bijvoorbeeld latere performanceproblemen in de keten te voorkomen.

De volgende testvormen komen in de functionele acceptatietest voor:

1. Functionaliteit.

Een allereerst uit te voeren test is of de functionaliteit overeenstemt met het ontwerp. Specifiek kan hier nog worden gekeken naar een ordentelijke foutafhandeling.

2. Regressietest.

Bij een regressietest wordt aandacht besteed aan de ongewijzigde onderdelen. Het is een berucht fenomeen bij testen: de wijzigingen zijn in orde, maar in productie blijkt een ongewijzigd onderdeel niet meer te functioneren. Dit komt vaak doordat de samenhang van systemen onvoldoende bekend is en daardoor de scope van de te testen onderdelen te nauw wordt genomen. Hierbij kan meespelen dat de aandacht bij de testers wat is verslapt, doordat er betrekkelijk weinig bevindingen worden geconstateerd en de test dus minder relevant lijkt te zijn. Tot slot kunnen de testers wat minder gemotiveerd zijn doordat ze bijvoorbeeld weinig interessante of uitdagende basisonderdelen moeten testen.

3. Integratie-/ketentest.

Bij een ketentest wordt het systeem in samenhang met andere systemen (de keten) getest. Daarbij kan het zowel in- als externe andere systemen betreffen. Het opzetten van deze test kan lastig zijn, omdat dat behoorlijke eisen stelt aan de testomgeving. Bij koppelingen met systemen van externe partijen moet idealiter de totale testomgeving zich tot en met die externe partijen uitstrekken. Is een dergelijke testomgeving niet op te tuigen, dan kunnen de interfaces met bepaalde systemen worden gesimuleerd. Simulatie blijft echter altijd een suboptimale oplossing omdat het uitgangspunt moet zijn dat de testomgeving zoveel mogelijk gelijk is aan de werkelijke productieomgeving.

Bijzonder aandachtspunten bij de integratie-/ketentest zijn de controle op input, output en beveiliging van het systeem en de datastromen. Komen geen data op de verkeerde plaats terecht, staan de data juist en tijdig als input ter beschikking of andersom: worden ze tijdig ter beschikking gesteld aan een ander systeem? Daarnaast kan er sprake zijn van functionele integratie, waarbij een bepaalde functionaliteit technisch door een combinatie van meerdere systemen wordt gerealiseerd. Vindt een wijziging plaats in een deelfunctionaliteit van een van de systemen, dan moet de gehele

MCTL – 6.4. Taakgebied Testen v1.19.1

functionaliteit getest worden. Een systeemdoorlooptest kan hier grote diensten bewijzen.

4. Beschikbaarheidstest.

Bij een beschikbaarheidstest wordt getest of een systeem langdurig goed blijft werken. Deze test doet denken aan de duurtests die worden uitgevoerd op fysieke producten om te zien of materialen langdurig tegen een bepaalde belasting bestand zijn.

5. Conversietest.

Indien van toepassing kan worden getest of de conversie goed verloopt. Een conversie kan zowel betrekking hebben op het converteren van data in hetzelfde systeem als het overzetten van data van een oud naar een nieuw systeem.

6. Documentatie.

Bij deze test wordt gekeken naar de wijzigingen in de voor het gebruik van het systeem bedoelde documentatie. Is deze documentatie correct, volledig en begrijpelijk?

7. Limiettest.

In de limiettest wordt gekeken naar kritische parameters en andere limieten waarbinnen het systeem moet functioneren. Ook het gedrag van het systeem zodra de limieten worden overschreden wordt getest. Doorgaans moeten hiervoor specifieke testgevallen worden gemaakt, omdat in de productiedatabase geen gevallen aanwezig horen te zijn die buiten de systeemlimieten vallen. Er kan dus niet gebruikgemaakt worden van testgevallen die uit productie zijn overgenomen.

8. Multi-user.

Een test of het systeem toegankelijk is voor meerdere personen tegelijkertijd zonder dat daardoor problemen zoals locking ontstaan.

Niet elke testvorm hoeft elke keer even relevant te zijn. Het is mogelijk enkele testvormen te combineren bij een relatief beperkte test of testvormen geheel weg te laten. Uiteraard moeten in de testfase alleen testvormen worden weggelaten die niet nodig zijn. Testvormen weglaten vanwege tijdgebrek is zeer onverstandig.

4. GEBRUIKERSACCEPTATIETEST

In het algemeen wordt de gebruikersacceptatietest uitgevoerd door gebruikers en key-users, met ondersteuning van functioneel specialisten. De omgeving waarin de test wordt uitgevoerd lijkt net zoals bij de functionele acceptatietest zoveel mogelijk op de uiteindelijke productiesituatie. Net zoals bij de functionele acceptatietest, dient eveneens een testplan te zijn opgesteld waarin de planning, de te testen onderdelen en een beschrijving van de testscripts en -data zijn opgenomen. Daarnaast dienen hier de acceptatiecriteria ter beschikking te staan.

Bij de gebruikersacceptatietest dient specifieke aandacht te zijn voor de happy flow en unhappy flow. Met andere woorden; er moet getest worden met testgevallen die feitelijk vrij standaard zijn (happy flow), maar daarnaast moeten ook de uitzonderlijke gevallen worden doorgelopen. Laatstgenoemde vereist nogal eens dat specifieke testgevallen

MCTL – 6.4. Taakgebied Testen v1.19.1

worden gedefinieerd. Vaak wordt door gebruikers bij testen gedacht aan werkelijk in productie aanwezige gevallen maar dan wordt doorgaans op zijn minst een deel van de unhappy flow gemist.

De gebruikersacceptatietest moet aantonen dat de uitgevoerde wijzigingen aan de wensen/eisen van de gebruikers voldoen. Er wordt dus getest ten opzichte van oorspronkelijke behoeften van de gebruikers. Deze behoeften zijn in Requirements management zo goed mogelijk gespecificeerd en daarna gevalideerd. Toch kan in die fase al veel zijn misgelopen. Het probleem van het testen ten opzichte van de oorspronkelijke behoeften is vaak dat die behoeften wat vaag en niet meetbaar of controleerbaar zijn beschreven.

Een voorbeeld ter illustratie:

Een manager wil de efficiency van een afdeling graag verhogen. Op de afdeling worden diverse administraties gevoerd; daarnaast is er een klein onderdeel 'advisering'. De gedachte is dat door meer inzet van computertechnologie in het onderdeel administratievoering, aldaar FTE's worden bespaard. Die kunnen vervolgens in het groeiende onderdeel 'advisering' worden ingezet. Het gevolg daarvan zal zijn dat met dezelfde personele bezetting een hogere output kan worden bereikt.

De administratieve processen worden grondig bekeken en al snel worden diverse oplossingen gedefinieerd, waarmee naar verwachting arbeidstijd kan worden bespaard. Te denken valt aan het wegwerken van de laatste papieren formulieren, het beter koppelen van diverse systemen, het verbeteren van veelgebruikte userinterfaces en het automatiseren van sommige handmatige controles. Een en ander wordt helder uitgewerkt in requirements en ontwerpdocumenten en zonder al te veel problemen gerealiseerd. Tijdens de functionele acceptatietest blijkt inderdaad dat alle onderdelen precies conform de functionele eisen zijn aangepast: de koppelingen functioneren, alle formulieren zijn gedigitaliseerd, userinterfaces zijn sterk verbeterd en op de aangegeven plaatsen zijn handmatige handelingen verdwenen.

Helaas blijkt bij het checken van de benodigde doorloop- en werktijd voor een gehele administratieve transactie, dat de gewenste tijdsbesparing (nog) niet wordt behaald. Het wijzigingstraject verloopt helemaal niet slecht en worden de wijzigingen op deze manier in productie genomen, dan zal er technisch en functioneel niet al te veel fout gaan. Maar het achterliggende doel: arbeidstijd besparen in de administratieve processen, wordt helaas niet behaald.

De testvormen die in de functionele acceptatietest zijn beschreven en ook bij de gebruikersacceptatietest voorkomen zijn de volgende.

1. functionaliteit;
2. beschikbaarheid;

MCTL – 6.4. Taakgebied Testen v1.19.1

3. conversietest;
4. documentatie;
5. keten- c.q. integratietest;
6. regressietest.

Aanvullende testvormen, specifiek voor de gebruikersacceptatietest, zijn:

1. Usability-test.

Bij de usability-test wordt het werkelijke gebruik van het systeem getest door een groep gebruikers. Er kan worden gekeken naar de gebruikersvriendelijkheid: schermopbouw, begrijpelijkheid, ondersteuning door aanvullende teksten, etc. Een dergelijke test kan worden uitgevoerd door gebruikers een bepaalde uit te voeren taak te geven en hen te verzoeken hardop alle gedachten uit te spreken bij elke handeling die zij verrichten. Ook wordt wel gebruikgemaakt van eye-trackingsoftware die volgt waarnaar iemand precies kijkt op het scherm. Het is mogelijk de tijd op te nemen zodat duidelijk wordt waar haperingen optreden en dus kennelijke onduidelijkheden zijn. Indien een gebruikersgroep niet direct beschikbaar is (bijvoorbeeld bij een website) kan voor 'remote usability testing' worden gekozen. Gebruikers op afstand wordt dan gevraagd deel te nemen. Een andere vorm is A/B-testen waarbij twee verschillende versies naast elkaar worden getest om te bepalen welke userinterface de beste resultaten geeft.

2. Load-/stresstest.

Bij een load-/stresstest wordt getest of het systeem bestand is tegen grote aantallen medewerkers, data en/of transacties. Natuurlijk moet hier een maximum zijn gedefinieerd en tot dat maximum moet het systeem conform verwachting functioneren. Daarboven moet het gecontroleerd stoppen of anderszins reageren (zoals een website die bij overschrijding van een maximumaantal bezoekers niet onderuit gaat, maar de bezoekers boven de limiet een boodschap voorschotelt). Probleem bij deze test is doorgaans dat deze in een gebruikersacceptatietestomgeving moeilijk uit te voeren is: het benodigde aantal gebruikers is niet voorhanden en grote aantallen transacties zijn moeilijk te forceren. Daarom wordt deze test in de praktijk veelal door infra-, applicatie-support en/of leverancier uitgevoerd waarna de daar genoteerde resultaten tijdens de gebruikersacceptatietest worden gecontroleerd.

3. Scenariotesten.

Hierbij wordt het bedrijfsproces via scenario's gesimuleerd. Aldus is te constateren of niet alleen het gewijzigde systeem, maar ook het bedrijfsproces conform verwachting functioneert.

Bij gebruikersacceptatietesten zijn de navolgende praktijkopmerkingen te maken.

Bij een gebruikersacceptatietest moeten eindgebruikers betrokken worden. Dat lijkt een voor de hand liggende zaak maar is in de praktijk niet altijd eenvoudig uitvoerbaar. De juiste gebruikers moeten betrokken worden, die vervolgens voldoende tijd ter beschikking moeten hebben. Onder 'juiste gebruikers' worden hier niet alleen de beste uit een gebruikersgroep verstaan, maar ook gebruikers

MCTL – 6.4. Taakgebied Testen v1.19.1

met een lager kennisniveau en minder computervaardigheden. Dat maakt het testen weliswaar lastiger, maar de zekerheid dat het systeem in productie zal doen wat het moet doen en zal worden gebruikt zoals het moet worden gebruikt, neemt vanzelfsprekend fors toe.

De faciliteiten waarmee gebruikers moeten testen dienen in orde te zijn. Niets is zo vervelend en frustrerend als gebrekkige faciliteiten of omgevingen die afwijken van de uiteindelijke productiesituatie. De verwarring die dat schept kan een test volledig frustreren en de aandacht wegzuigen van waar de test werkelijk om draait.

De ondersteuning vanuit functioneel specialisten is nodig. Dat betekent niet dat functioneel specialisten de test overnemen. Bij problemen met de testomgeving kan een functioneel specialist echter doorgaans gemakkelijker ingrijpen en samen met infra-, applicatiesupport en leverancier het een en ander oplossen. Ook bij constatering van bevindingen kan een functioneel specialist een oogje in het zeil houden: zijn het echte bevindingen of worden zij bijvoorbeeld veroorzaakt door een (oude) werkwijze die niet meer van toepassing is?

Het is goed ernaar te streven dat tijdens een gebruikersacceptatietest niet te weinig, maar ook niet te veel bevindingen worden geconstateerd. Indien tijdens eerdere tests zo grondig is getest, dat tijdens de gebruikersacceptatietests vrijwel niets meer wordt gevonden, dan zal de gebruikersgroep in reactie daarop het testen als zinloos gaan ervaren. Is anderzijds het aantal bevindingen bij de gebruikers tijdens het testen erg hoog, dan dreigt juist een negatieve sfeer te ontstaan. Gebruikers zullen gaan denken dat het niets is en ook niets zal worden. Omdat testers daarnaast hun gewone werk doen, kan die negativiteit op de werkvloer gaan rondzingen. In feite moet het aantal bevindingen tijdens de gebruikersacceptatietest worden gedoseerd. Lijkt het er vlak na de start van de gebruikersacceptatietest al op dat het aantal bevindingen geheel uit de hand zal lopen, dan kan de gebruikersacceptatietest eventueel worden afgebroken. Vervolgens worden eerst voorgaande tests beter uitgevoerd en bevindingen opgelost, waarna de gebruikersacceptatietest kan worden herstart.

5. PRODUCTIEACCEPTATIETEST

De productieacceptatietest is een door (een vertegenwoordiging van) de toekomstige beheerders van de productieomgeving uit te voeren test. Ook deze test wordt uitgevoerd in een omgeving die zoveel mogelijk op die productieomgeving lijkt. De test moet aantonen dat de uitgevoerde wijzigingen aan de wensen/eisen vanuit beheer voldoen. De nadruk bij een productieacceptatietest ligt in het algemeen sterk op de technische kant. Toch verdient deze test op twee manieren aandacht van de functioneel specialisten.

Ten eerste dienen functioneel specialisten zich af te vragen en te testen of het systeem functioneel te beheren is. Ontbrekende bijgewerkte handleidingen, rammelende workflows of niet bijgewerkte helpteksten zijn wellicht voor gebruikers nog wel overkomelijk, maar functioneel specialisten moeten daar vanuit de optiek van het leveren van ondersteuning eveneens een oordeel over vellen. Bovendien houden functioneel specialisten intern vaak documenten en tools bij, die gebruikt worden ter ondersteuning. Zijn die inderdaad bijgewerkt en kunnen ze ingezet worden nadat bij gebruikers het gewijzigde systeem in productie is gekomen?

Ten tweede kunnen aan de technische zijde tijdens de productieacceptatietest bevindingen worden gedaan en daaropvolgend beslissingen worden genomen die functionele impact hebben. In dit soort situaties is op zijn minst overleg tussen functioneel, infra- en applicatiesupport nodig en moet er gezamenlijk naar de bevindingen worden gekeken. Hoewel technische perikelen roet in het eten kunnen gooien en deze testfase vlak voor de inproductie ligt – met alle tijdsdruk van dien – moeten nu de belangen van de gebruikers blijven prevaleren. Worden hier echte 'showstoppers' geconstateerd, dan kan het systeem toch niet in productie worden genomen, ondanks het feit dat de gebruikersacceptatietest en functionele acceptatietest goede resultaten hebben opgeleverd.

De testvormen specifiek voor de productieacceptatietest zijn:

1. Back-up & restoretest.

Het testen of de back-up en restore nog goed functioneren. De test wordt uitgevoerd door infrasupport en door functioneel specialisten wordt daarop een inhoudelijke controle uitgeoefend.

2. Herstartbaarheid.

Is het systeem herstartbaar na (onverwachte) uitval? Uit te voeren door infrasupport. Functioneel is er aandacht voor consistentie in data, dataverlies en juiste foutafhandeling.

3. Installatie-/de-installatietest.

Hierbij wordt functioneel gekeken naar de volledigheid en juistheid van de installatie (is niets vergeten, komen alle updates bij precies de juiste gebruikersgroep terecht?) en de-installatie (komt het oude systeem precies zoals het was weer in de lucht, kunnen we met het oude systeem doorwerken?).

4. Load-/stresstest.

Zie voor toelichting de hiervoor beschreven gebruikersacceptatietest.

5. Performance.

Voldoet het systeem aan de performance-eisen?

6. Uitwijktest.

Functioneert uitwijk nog conform de afspraken? Ook hier is vooral aandacht voor de functionele aspecten: komt het gewijzigde systeem in geval van uitwijk binnen de gestelde tijd weer beschikbaar, zijn er geen negatieve gevolgen voor andere systemen, is het inderdaad mogelijk in uitwijksituatie met het systeem te werken, is inwijk (terugkeren naar de gebruikelijke werksituatie) nog steeds mogelijk?

Sommige testvormen komen hier voor, terwijl ze ook al bij eerdere testsoorten zijn genoemd. In de PAT ligt de focus vanzelfsprekend anders dan in de FAT of de GAT.

6. OPSTELLEN VRIJGAVE-ADVIES/VERBETERPUNTEN

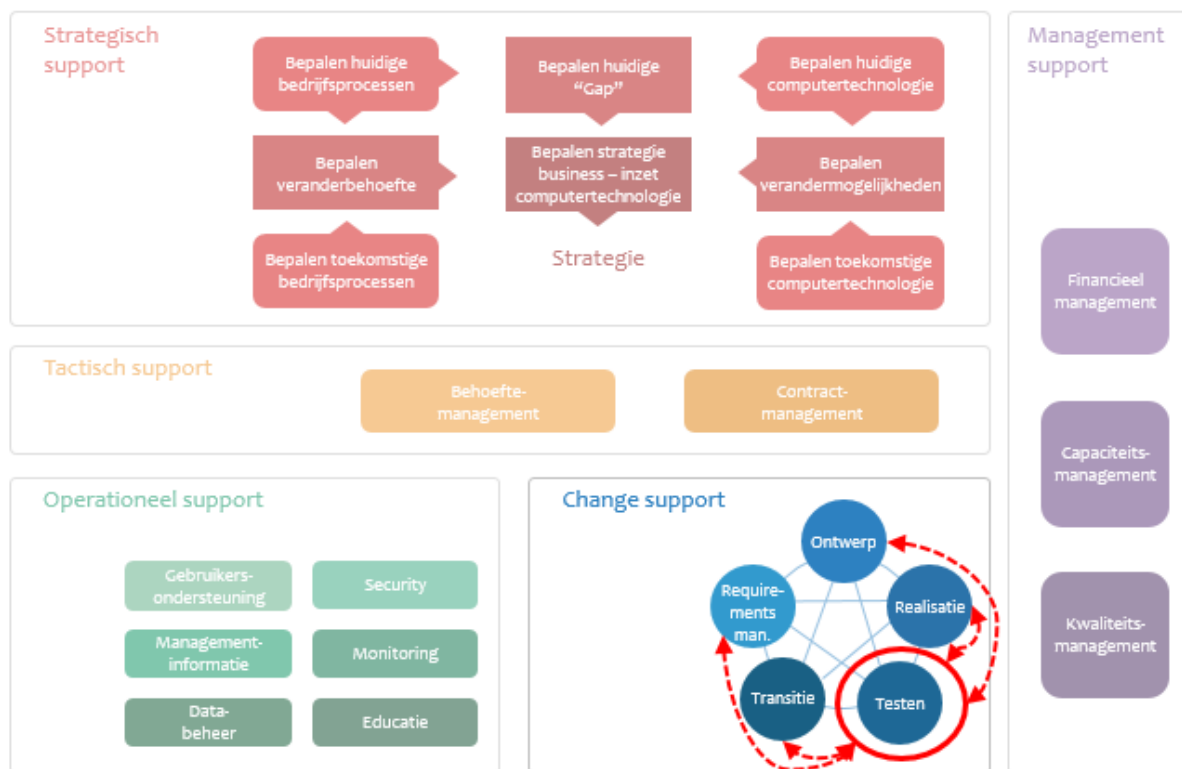
Nadat alle tests afgerond zijn, wordt een vrijgave-advies opgesteld. Dat betekent niet dat op dat moment alle geconstateerde bevindingen opgelost moeten zijn. Ondanks het feit dat er nog openstaande bevindingen zijn, kan ervoor worden gekozen toch te adviseren de wijzigingen in productie te nemen. In het advies worden de eventueel nog openstaande bevindingen vermeld, inclusief de consequenties daarvan voor de productieomgeving. Ook kunnen in het vrijgave-advies afspraken over later op te lossen bevindingen worden vermeld.

Tevens kunnen op dit moment de in het hele toets- en testtraject geconstateerde verbeterpunten bij elkaar gebracht worden en gedocumenteerd voor later gebruik. Ook indien het toets- en testtraject naar volle tevredenheid is verlopen, kunnen er nog verbeterpunten zijn. Toch is een waarschuwing wel op zijn plaats: er moet hier niet worden gestreefd naar het perfecte maar naar het optimale. Die twee begrippen zijn veelal niet gelijk aan elkaar.

RELATIES MET ANDERE ONDERDELEN VAN MCTL

Dit taakgebied kent de volgende belangrijke relaties.

MCTL – 6.4. Taakgebied Testen v1.19.1



De belangrijkste relaties zijn die met de taakgebieden **Realisatie** en **Transitie**. Vanuit eerstgenoemde zullen de bijgewerkte objecten worden aangeleverd, die het voornaamste onderwerp van de test zullen zijn. Na goedkeuring zullen alle geteste onderdelen aan Transitie worden opgeleverd. Andere te noemen relaties zijn die met **Requirements management** en **Ontwerp**, omdat de uitgewerkte requirements en bijgewerkte ontwerpen op diverse plaatsen binnen Testen worden gebruikt.

OPMERKINGEN

De navolgende opmerkingen zijn betreffende dit taakgebied te maken.

1. TESTSOORTEN EN -VORMEN

In de wereld van het testen worden veel termen door elkaar gebruikt. Hieronder worden de testsoorten en -vormen die in TMap worden beschreven opgesomd. Testvormen kunnen binnen de verschillende testsoorten worden gebruikt. Doorgaans is een bepaalde testvorm daarom binnen meerdere testsoorten terug te vinden.

De navolgende testsoorten zijn te onderscheiden.

- Unittest (UT): Test door de ontwikkelaar in een ontwikkelomgeving. Check op technisch goed functioneren van een unit (individuele wijziging).

MCTL – 6.4. Taakgebied Testen v1.19.1

- Unitintegratietest (UIT): Test door de ontwikkelaar in een ontwikkelomgeving. Check op technisch goed functioneren van een logische groep units (aantal wijzigingen, bijvoorbeeld in een release).
- Systeemtest (ST): Test van het systeem door de leverancier in een laboratoriumomgeving t.o.v. de (niet-)functionele specificaties en het technisch ontwerp.
- Systeemintegratietest (SIT): Een door de toekomstige gebruiker uitgevoerde test op de systeeminterfaces in een zoveel mogelijk op de productieomgeving lijkende omgeving.
- Functionele acceptatietest (FAT): Een door functioneel specialisten uitgevoerde test in een zoveel mogelijk op de productieomgeving lijkende omgeving. Deze test moet aantonen dat de uitgevoerde wijzigingen aan de functionele eisen voldoen.
- Gebruikersacceptatietest (GAT): Een door (een vertegenwoordiging van) de toekomstige gebruikers uitgevoerde test in een zoveel mogelijk op de productieomgeving lijkende omgeving. Deze test moet aantonen dat de uitgevoerde wijzigingen aan de wensen/eisen van de gebruikers voldoen.
- Productieacceptatietest (PAT): Een door (een vertegenwoordiging van) de toekomstige beheerders uitgevoerde test in een zoveel mogelijk op de productieomgeving lijkende omgeving. Deze test moet aantonen dat de uitgevoerde wijzigingen aan de wensen/eisen vanuit beheer voldoen. Beheer kan hierbij zowel de technische als functionele kant betreffen, hoewel de nadruk doorgaans sterk op de technische kant ligt.

De meest voorkomende testvormen zijn de navolgende.

- Back-up & restoretest: Testen of back-upproces correct functioneert.
- Beschikbaarheid: Blijft het systeem langdurig correct werken?
- Conversietest: Test op conversie van data.
- Documentatie: Controle op volledigheid, juistheid, begrijpelijkheid.
- Functionaliteit: Is de functionaliteit overeenkomstig het ontwerp?
- Hackers-/penetratietest: Test op de mogelijkheid om het systeem van buitenaf oneigenlijk te gebruiken.
- Herstartbaarheid: Is het systeem bij onverwacht afbreken conform afspraken herstartbaar?
- Installatie-/de-installatietest: Test of de installatie en de-installatie van de wijzigingen werken conform afspraken.
- Interface-/ketentests: Tests rondom de aansluiting van het systeem op andere systemen (technisch en functioneel).
- Limiettest: Test op gedefinieerde limieten (functioneel, bijvoorbeeld maximumsalaris).
- Load-/stresstest: Test op het verwerken van grote hoeveelheden data, transacties, gebruikers, klanten en/of bezoekers.
- Middelengebruik: Test op het gebruik van resources.
- Monkeytest: Test op de bestandheid tegen onverwachte handelingen (foolproof).
- Multi-user: Kunnen meerdere/kan het afgesproken aantal gebruikers van het systeem gebruik maken?

MCTL – 6.4. Taakgebied Testen v1.19.1

- Performance: Voldoet het systeem aan de performance-eisen?
- Regressie: Testen op het functioneren van het geïntegreerde geheel, met de focus op de ongewijzigde onderdelen.
- Rollback: Test of een rollback werkt conform afspraken.
- Scenariotesten: Testen via scenario's waarin bedrijfsprocessen worden gesimuleerd.
- Uitwijktest: Test of uitwijk conform afspraken functioneert.
- Usability: Test op gebruikersvriendelijkheid, de juistheid van het functioneren van een userinterface, de opbouw van een userinterface (conform afspraken?).

2. TESTEN MET PRODUCTIEDATA

Of het testen met productiedata aan te bevelen of juist af te raden is, roept al lang veel discussie op. Bij een dergelijke test wordt de gehele productiedatabase overgezet naar een acceptatietestomgeving, waarna de test plaatsvindt. De voordelen zijn onder andere dat het zo geen tijd kost om een bestand met testgevallen op te bouwen en dat de werkelijke productiesituatie heel goed wordt nagebootst. Toch zijn er de nodige nadelen te benoemen. Een productiedatabase kan heel groot zijn, waardoor de acceptatietestomgeving net zo omvangrijk wordt en daarmee onhandelbaarder. Sommige organisaties hanteren een meer principieel standpunt: productiedata worden beschermd tegen 'verkeerde ogen'. In testomgevingen zijn dergelijke waarborgen er vaak niet. Hierdoor kan het gebeuren dat productiedata toch beschikbaar komen bij personen die daar geen toegang toe zouden mogen hebben (tot zelfs extern bij leveranciers). Nu is het mogelijk productiedata te scambelen, maar dat heeft weer als nadeel dat bepaalde functionele tests niet meer uitgevoerd kunnen worden. Zijn bijvoorbeeld alle namen gescrambeld, dan wordt het onmogelijk een functionaliteit als 'zoek verwanten op naam' te gebruiken. Tot slot is het vanuit functioneel oogpunt belangrijk dat niet alle te testen gevallen werkelijk in de productiedatabase hoeven voor te komen. Stel dat er in de loonadministratie een controle zit op overschrijding van het maximumsalaris, hoe groot is dan de kans dat in de productiedatabase iemand is opgenomen die precies op dat maximum zit? Of, als die controle getest moet worden, een persoon met een te hoog salaris in de productiedatabase aanwezig is? Als het goed is, komen dergelijke gevallen niet in de productiedatabase voor, maar om de controle te testen, zou een dergelijk geval wel in een testset opgenomen moeten zijn.

Het kan zinvol zijn met productiedata te testen. Daarbij kan nog worden gekozen voor het testen met de volledige productieset of met een extract daarvan (bijvoorbeeld 1 op de 100 random selecteren en overzetten naar een testomgeving). Toch is het niet de enige test die uitgevoerd zou moeten worden; het samenstellen van een of meerdere testsets met eigen, specifieke testgevallen blijft noodzaak.

3. TESTEN IN PRODUCTIE

Een absolute 'no go' voor veel bedrijven is testen in productie. Is vasthouden aan dat principe altijd haalbaar? Er zijn genoeg situaties te bedenken waarin geen representatieve testomgeving op te bouwen is. Denk eens aan een productielijn voor veevoer, bier of

auto's. Het is volstrekt ondenkbaar een dergelijk productieproces geheel in een testomgeving op te bouwen. Weliswaar is een simulatie uit te voeren, maar uiteindelijk zal pas in productie blijken of een en ander geheel correct is. Juist functioneel is het dan noodzaak op een of andere manier de gevolgen van eventuele productiefouten te beperken. Middels pre-productieruns kunnen eventuele fouten nog worden opgespoord. In productiebedrijven is het overigens niet vreemd om, voordat een productielijn volledig productie gaat draaien, een inlooperperiode te hebben die soms maanden in beslag kan nemen.

4. TESTEN IN PRODUCTIE-LOOKALIKE-OMGEVINGEN

In de functionele acceptatietest en gebruikersacceptatietest is het de bedoeling in een omgeving te testen die zoveel mogelijk gelijk is aan de uiteindelijke productieomgeving. Dat brengt het gevaar met zich mee, dat functioneel specialisten en de gebruikers die de tests uitvoeren in de war raken. Dit wordt mede veroorzaakt doordat deze mensen geneigd zijn om het gewijzigde systeem in de testomgeving te vergelijken met de werkelijke, huidige productiesituatie. Door maar genoeg schermen open te hebben staan, kan de test worden aangezien voor productie en andersom. Er moet dus altijd duidelijk blijven welke omgeving de testomgeving is. Dat kan onder andere door de fysieke inrichting: bepaalde apparaten/ruimtes geven altijd uitsluitend toegang tot de testomgeving. Ook kan met behulp van een indicator (kleurverschil, balk met het woord 'test') op elk scherm aangegeven worden in welke omgeving men zich bevindt.

Tot slot nog een aardige anekdote hierover.

Een fabriek maakt apparaten die tot wel enige miljoenen euro's per stuk kunnen kosten. Bij een aanpassing van het ordersysteem wordt in de acceptatietestomgeving een nieuwe order ingevoerd en goedgekeurd. Vanwege allerlei koppelingen en dashboards gaat bij de directie al snel de champagne open: dergelijke orders komen tenslotte niet elke dag voor. Pas na enige tijd wordt duidelijk dat het helaas geen echte order betreft.

5. TESTEN WEINIG GEBRUIKTE FUNCTIONALITEIT

Systemen bevatten doorgaans functionaliteit die veel en functionaliteit die weinig wordt gebruikt. Een voorbeeld van het laatste is een onderdeel dat alleen wordt gebruikt voor de jaarafsluiting. Bij testen worden weinig gebruikte systeemonderdelen nogal eens veronachtzaamd. Dit kan een bewuste strategie zijn; het is goed mogelijk sommige functionaliteit alleen te testen vlak voor het gebruikt wordt. In het genoemde geval van de jaarafsluiting in januari wordt die functionaliteit in december getest. De rest van het jaar gebeurt dat niet, ook niet als er wijzigingen worden aangebracht die invloed hebben op dat onderdeel. Het is een kwestie van een keuze maken tussen:

- Alle onderdelen van het systeem worden in zijn geheel altijd up-to-date gehouden

MCTL – 6.4. Taakgebied Testen v1.19.1

- Sommige systeemonderdelen zullen feitelijk in een bepaalde periode misschien niet werken, maar worden getest en productieklaar gemaakt vlak voor het gebruik in productie weer een aanvang neemt.

De schoning

Een systeem draait eigenlijk al jaren vrijwel probleemloos en kleinere updates worden vlot gerealiseerd. Plots, na vijf jaar, ontstaan in productie grote problemen. Data verdwijnt spoorloos en het regent foutmeldingen. Na enig speurwerk blijkt dat bij de initiële bouw van het systeem ook een module is gebouwd die ervoor zorgt dat data ouder dan vijf jaar automatisch wordt verwijderd. De oorspronkelijke ontwerpers en bouwers waren al lang van het toneel verdwenen en bij tussentijdse updates was de betreffende module steeds overgeslagen. Bij tussentijdse overdracht was de aandacht voor de module steeds verder verslapt tot het tot het nulpunt was gedaald. Na vijf jaar was er opeens toch weer aandacht voor de, zo bleek, werkelijk nuttige maar tot dan toe "slapende" functionaliteit.

CERTIFICERING/PROEFEXAMENVRAGEN

Voor MCTL kunt u zich certificeren op *foundation*, *advanced* en *expert* niveau. Het foundationniveau toetst uw kennis van MCTL. Het advanced en expert level toetsen uw vaardigheid in het toepassen van MCTL. In een apart onderdeel, 'MCTL Certificering', vindt u alle informatie over de drie niveaus. Hierna vindt u proefexamenvragen op foundation- en advanced-basisniveau.

1. MCTL FOUNDATION - PROEFEXAMENVRAGEN

Voor dit hoofdstuk zijn de volgende proefexamenvragen beschikbaar. Maak deze zonder terug te bladeren. De uitleg vindt u direct hierna.

16-1. Binnen het taakgebied Testen wordt onderscheid gemaakt tussen toetsen en testen. Wat wordt onder toetsen verstaan, en wat onder testen?

- a. Bij testen wordt de gebruikersacceptatietest benoemd, bij toetsen de productieacceptatietest.
- b. Bij testen wordt de gebruikersacceptatietest benoemd, bij toetsen de reviews.
- c. Bij testen wordt de integratietest benoemd, bij toetsen de reviews.
- d. Het zijn twee termen waaronder echter dezelfde tests (checkmomenten) vallen.

16-2. Binnen het taakgebied Testen wordt onderscheid gemaakt tussen toetsen en testen. Wat is het verschil tussen beide?

- a. Er is geen verschil, het zijn synoniemen.
- b. Bij toetsen worden eindproducten gecontroleerd, bij testen worden tussenproducten gecontroleerd.
- c. Bij toetsen worden tussenproducten gecontroleerd, bij testen worden eindproducten gecontroleerd.

- d. Bij toetsen wordt vooraf gecontroleerd of er wel getest kan gaan worden, dus of aan de voorwaarden is voldaan. Daarna wordt er getest.

16-3. Acceptatiecriteria zijn meetbare condities waaraan moet worden voldaan. Deze criteria kunnen op meerdere plaatsen worden gebruikt, onder andere bij Testen. Wat is de relatie tussen Testen en acceptatiecriteria?

- a. Binnen Testen worden de acceptatiecriteria gebruikt.
- b. Binnen Testen worden de acceptatiecriteria gebruikt, maar indien nodig ook aangepast.
- c. Binnen Testen worden de acceptatiecriteria opgesteld.
- d. Binnen Testen worden de acceptatiecriteria aangepast.

16-4. Binnen Testen wordt onder andere een gebruikersacceptatietest uitgevoerd. Deze heeft tot doel te bepalen of:

- a. wat gebruikers hebben gevraagd ook goed in de specificaties is geformuleerd, zodat er geen fouten bij de bouw worden gemaakt.
- b. de uitgewerkte requirements juist zijn uitgevoerd.
- c. de inproductie name volgens afspraak kan verlopen.
- d. de uitgevoerde wijzigingen aan de wensen/eisen van de gebruikers voldoen.

16-5. Inspectie wordt als werkwijze in MCTL gebruikt binnen:

- a. Testen.
- b. Requirements management.
- c. Ontwerp.
- d. Gebruikersondersteuning.

16-6. Testen heeft relaties met andere taakgebieden binnen taakcluster Change support. Totaal betreft dit:

- a. Realisatie als inputbron en Transitie om het goedgekeurde resultaat aan op te leveren.
- b. Requirements management als inputbron, Transitie om het goedgekeurde resultaat aan op te leveren.
- c. Ontwerp als inputbron.
- d. Requirements management, Ontwerp en Realisatie als inputbronnen en Transitie als taakgebied om het goedgekeurde resultaat in gebruik te laten nemen.

2. MCTL FOUNDATION –PROEFEXAMENVRAGEN MET ANTWOORDEN EN UITLEG

Hierna vindt u de proefexamenvragen direct daarachter de antwoorden en uitleg.

16-1. Binnen het taakgebied Testen wordt onderscheid gemaakt tussen toetsen en testen. Wat wordt onder toetsen verstaan, en wat onder testen?

- a. Bij testen wordt de gebruikersacceptatietest benoemd, bij toetsen de productieacceptatietest.
- b. Bij testen wordt de gebruikersacceptatietest benoemd, bij toetsen de reviews.
- c. Bij testen wordt de integratietest benoemd, bij toetsen de reviews.
- d. Het zijn twee termen waaronder echter dezelfde tests (checkmomenten) vallen.

MCTL – 6.4. Taakgebied Testen v1.19.1

- a. Onjuist. De gebruikersacceptatietest hoort inderdaad bij testen, maar de productieacceptatietest hoort daar eveneens bij.
- b. Juist. Dit zijn voorbeelden die bij testen c.q. toetsen worden genoemd. Zie hoofdstuk 16.
- c. Onjuist. De integratietest wordt niet bij testen benoemd.
- d. Onjuist. Testen is het controleren van de diverse eindproducten, terwijl met toetsen de diverse tussenproducten kunnen worden beoordeeld.

16-2. Binnen het taakgebied Testen wordt onderscheid gemaakt tussen toetsen en testen. Wat is het verschil tussen beide?

- a. Er is geen verschil, het zijn synoniemen.
 - b. Bij toetsen worden eindproducten gecontroleerd, bij testen worden tussenproducten gecontroleerd.
 - c. Bij toetsen worden tussenproducten gecontroleerd, bij testen worden eindproducten gecontroleerd.
 - d. Bij toetsen wordt vooraf gecontroleerd of er wel getest kan gaan worden, dus of aan de voorwaarden is voldaan. Daarna wordt er getest.
-
- a. Onjuist. Indien er geen verschil zou zijn, zou het wel heel onhandig zijn twee termen te gebruiken.
 - b. Onjuist. Het is precies andersom: bij toetsen gaat het om tussenproducten, bij testen om eindproducten.
 - c. Juist. Dit is precies het verschil tussen toetsen en testen. Zie hoofdstuk 16.
 - d. Onjuist. Hoewel een check vooraf op alle randvoorwaarden aan te raden is, is dit niet de essentie van toetsen.

16-3. Acceptatiecriteria zijn meetbare condities waaraan moet worden voldaan. Deze criteria kunnen op meerdere plaatsen worden gebruikt, onder andere bij Testen. Wat is de relatie tussen Testen en acceptatiecriteria?

- a. Binnen Testen worden de acceptatiecriteria gebruikt.
 - b. Binnen Testen worden de acceptatiecriteria gebruikt, maar indien nodig ook aangepast.
 - c. Binnen Testen worden de acceptatiecriteria opgesteld.
 - d. Binnen Testen worden de acceptatiecriteria aangepast.
-
- a. Juist. Acceptatiecriteria worden in taakgebied Kwaliteitsmanagement opgesteld en eventueel aangepast (afgeleid van kwaliteitsnormen). In Testen kunnen ze vervolgens worden gebruikt. Zie hoofdstuk 16.
 - b. Onjuist. Acceptatiecriteria worden in taakgebied Testen gebruikt. Mocht er iets aan de acceptatiecriteria moeten veranderen, dan wordt dat teruggekoppeld met Kwaliteitsmanagement en aldaar eventueel aangepast.
 - c. Onjuist. Acceptatiecriteria worden in taakgebied Kwaliteitsmanagement opgesteld en eventueel aangepast. In Testen worden ze gebruikt.
 - d. Onjuist. Acceptatiecriteria worden in taakgebied Kwaliteitsmanagement opgesteld en eventueel aangepast. In Testen worden ze gebruikt.

16-4. Binnen Testen wordt onder andere een gebruikersacceptatietest uitgevoerd. Deze heeft tot doel te bepalen of:

- a. wat gebruikers hebben gevraagd ook goed in de specificaties is geformuleerd, zodat er geen fouten bij de bouw worden gemaakt.
- b. de uitgewerkte requirements juist zijn uitgevoerd.

MCTL – 6.4. Taakgebied Testen v1.19.1

- c. de inproductiename volgens afspraak kan verlopen.
 - d. de uitgevoerde wijzigingen aan de wensen/eisen van de gebruikers voldoen.
-
- a. Onjuist. Dit kan onderdeel zijn van toetsen.
 - b. Onjuist. Uiteraard moet ook worden gecontroleerd of de wijzigingen conform requirements zijn uitgevoerd, maar uiteindelijk gaat het erom of voldaan is aan de wensen/eisen van de gebruikers. Dit laatste staat in de gebruikersacceptatietest centraal.
 - c. Onjuist. Uiteraard is dit ook een onderdeel van testen, maar in de gebruikersacceptatietest staat centraal of voldaan is aan de wensen/eisen van gebruikers.
 - d. Juist. Uiteindelijk gaat het erom in een testsituatie te constateren of voldaan is aan de wensen/eisen van gebruikers. Dit staat in de gebruikersacceptatietest centraal. Zie hoofdstuk 16.

16-5. Inspectie wordt als werkwijze in MCTL gebruikt binnen:

- a. Testen.
 - b. Requirements management.
 - c. Ontwerp.
 - d. Gebruikersondersteuning.
-
- a. Juist. Het is een werkwijze die wordt benoemd in toetsen binnen taakgebied Testen. Zie hoofdstuk 16.
 - b. Onjuist. Inspectie wordt benoemd in taakgebied Testen.
 - c. Onjuist. Inspectie wordt benoemd in taakgebied Testen
 - d. Onjuist. Inspectie wordt benoemd in taakgebied Testen.

16-6. Testen heeft relaties met andere taakgebieden binnen taakcluster Change support. Totaal betreft dit:

- a. Realisatie als inputbron en Transitie om het goedgekeurde resultaat aan op te leveren.
 - b. Requirements management als inputbron, Transitie om het goedgekeurde resultaat aan op te leveren.
 - c. Ontwerp als inputbron.
 - d. Requirements management, Ontwerp en Realisatie als inputbronnen en Transitie als taakgebied om het goedgekeurde resultaat in gebruik te laten nemen.
-
- a. Onjuist. Dit zijn inderdaad twee relaties, maar er bestaan er meer.
 - b. Onjuist. Dit zijn inderdaad twee relaties, maar er bestaan er meer.
 - c. Onjuist. Dit is inderdaad een relatie, maar er bestaan er meer.
 - d. Juist. Dit zijn inderdaad alle relaties die Testen heeft binnen taakcluster Change support. Zie hoofdstuk 16.

3. MCTL ADVANCED-BASIS - PROEFEXAMENVRAGEN

Voor dit hoofdstuk zijn de volgende proefexamen vragen op advanced-basisniveau beschikbaar. Het zijn open vragen waarop u de antwoorden in de tekst van dit hoofdstuk kunt terugvinden. Om veel herhaling te voorkomen is daarom hier geen aparte uitleg per vraag opgenomen.

Vraag 1 (5 punten): Leg het verschil uit tussen toetsen en testen.

MCTL – 6.4. Taakgebied Testen v1.19.1

Vraag 2 (5 punten): Noem twee technieken om te toetsen.

Vraag 3 (5 punten): Binnen taakgebied Testen vinden een functionele en gebruikersacceptatietest plaats. Voordat deze plaatsvinden hebben een leverancier en infra- en applicatiesupport ook getest. Het is aan te bevelen een controle op deze tests uit te voeren. Stel dat de leverancier niet goed genoeg heeft getest. Waar zal dat blijken in taakgebied Testen?

Vraag 4 (5 punten): In de productieacceptatietest kan bijvoorbeeld de herstartbaarheid worden getest. Wordt dit onderdeel van de productieacceptatietest ook door functioneel specialisten uitgevoerd?

Vraag 5 (5 punten): Ziet MCTL testen in een productieomgeving als principieel onjuist?

Vraag 6 (5 punten): Noem drie overwegingen bij het testen met behulp van een kopie van data uit de productieomgeving. Deze overwegingen mogen zowel positief als negatief zijn.

NUTTIGE WEBSITES EN BOEKEN

Vanuit functioneel perspectief zijn de volgende websites interessant voor taakgebied Testen.

- www.mctl.nl
MCTL.nl – Website met alle informatie over MCTL; de achtergrond, een beschrijving van het model, video's, artikelen, etc. etc. Alle documenten, waaronder dit document, zijn vanaf deze website te downloaden.
- www.bisl.nl
BiSL.nl – Website met alle informatie over BiSL. BiSL is, als voorganger van MCTL, interessant vanwege de verzameling Best Practices, whitepapers en artikelen die op deze website zijn te vinden.

Vanuit functioneel perspectief zijn de volgende boeken interessant voor taakgebied Testen.

- Aalst, L. van der, Baarda, R., Roodenrijs, E., Vink, J. & Visser, B. (2008). *TMap Next Business Driven Test Management*. 's-Hertogenbosch: Tutein Nothenius.
- Aalst, L. van der, Broekman, B., Koomen, T. & Vroon, M. (2006). *TMap Next voor resultaatgericht testen*. 's-Hertogenbosch: Tutein Nothenius.
- Best, B. de (2006). *Acceptatiecriteria*. Den Haag: Sdu.
- Best, B. de (2009). *Acceptatiecriteria*. Lierderholthuis: Leonon Media.
- Black, R., Veenendaal, E. van & Graham, D. (2012). *Foundations of software testing*. Hampshire: Cengage Learning.
- Bouman, E. (2004). *SmartTEST*. Den Haag: Ten Hagen & Stam.
- Cannegieter, J. J., Veenendaal, E. van, Vliet, E. van der & Zwan, M. van der (2008). *Reviews in de praktijk*. Den Haag: Sdu.
- CMG. (1999). *Testframe*. Den Haag: Ten Hagen & Stam.

MCTL – 6.4. Taakgebied Testen v1.19.1

- Dekkers, M. & Schaaf, B. van der (2001). *Leidraad voor testen e-business*. Delft: Eburon Publishers.
- Spillner, A., Linz, T. & Pol, M. (2004). *Testen volgens ISEB*. 's-Hertogenbosch: Tutein Nothenius.
- Stam, A. van & 't Hek, P. van (2014). *Acceptatietesten in de praktijk*. Den Haag: BIM Media.