



DE MOGELIJKHEDEN VOOR IT-AUDITORS

Beoordelen **onderhoudbaarheid** maatwerkapplicaties

IT-auditors zijn gewend om IT-assurance opdrachten uit te voeren met betrekking tot beheersing en beveiliging van informatietechnologie. Wat te doen als een oordeel wordt gevraagd over de onderhoudbaarheid van maatwerkapplicaties? Achmea Internal Audit heeft samen met de Software Improvement Group (SIG) een onderzoek gedaan, gericht op het geven van een dergelijk oordeel. Met dit artikel willen wij duidelijk maken dat software-ontwikkeling ook voor IT-auditors een begaanbaar gebied is. Auditors kunnen tot een onderbouwd oordeel komen over de onderhoudbaarheid van maatwerkapplicaties en adviseren over het ontwikkelproces om onderhoudbaarheid van applicaties te verbeteren.

MARK VAN DER KRIFT EN MARIJN DESSENS

Het vakgebied IT-auditing is ontstaan vanuit de accountancy, met het doel accountants te ondersteunen bij het beoordelen van geautomatiseerde informatievoorziening. Het vak is inmiddels geëvolueerd tot het beoordelen van en het adviseren over alle aspecten van informatietechnologie in een organisatie. Wat nu als aan u gevraagd wordt een oordeel te geven over de werkzaamheden van de afdeling die verantwoordelijk is voor de ontwikkeling van maatwerkapplicaties? Opgeleverde functionaliteit kunnen we meten. Maar kunnen we applicaties ook effectief en efficiënt aanpassen om snel nieuwe proposities in de markt te zetten en in te spelen op veranderende wet- en regelgeving?

Effectief en efficiënt aanpassen van applicaties vereist onderhoudbare applicaties. Hoe beoordelen wij de onderhoudbaarheid? In dit artikel staan handvatten om te komen tot een objectief en onderbouwd oordeel over onderhoudbaarheid. Met dit resultaat in de hand kunnen wij vervolgens adviseren over het meten en waarborgen van onderhoudbaarheid in software-ontwikkelprocessen.

ACHTERGROND ACHMEA

Achmea is met bijna 18.000 medewerkers en vele miljoenen klanten de grootste verzekeraar van Nederland. Het concern heeft kantoren op negen kernlocaties in Nederland. Achmea is voortgekomen uit het samengaan van meerdere onderlinge verzekeraars, bijvoorbeeld de fusie met Interpolis

in 2005 en met Agis Zorgverzekeringen in 2007. Meer recent is de fusie met De Friesland Zorgverzekeraar. Achmea biedt producten op het gebied van schade-, zorg-, inkomens-, pensioen- en levensverzekeringen. De verzekeringen worden in Nederland en Europa verkocht aan bedrijven en particulieren via een aantal merken, waaronder Interpolis, Centraal Beheer en FBTO. Achmea heeft een coöperatieve achtergrond waarbij de belangen van klanten, medewerkers, partners en aandeelhouders in balans zijn. De ontwikkeling en het beheer van IT-voorzieningen zijn binnen Achmea gecentraliseerd in één bedrijfsafdeling, IM&IT, waarin 2.500 medewerkers werkzaam zijn.

Het IT-landschap van Achmea is een afspiegeling van de geschiedenis van het bedrijf: door fusies, overnames en veranderingen in de organisatie is in de loop van de tijd een vrij complex landschap ontstaan van vele applicaties gebaseerd op verschillende technologieën. Door de voortdurende veranderingen, zowel door fusies als een veranderende organisatie van Achmea zelf, neemt de dynamiek in het IT-landschap in de nabije toekomst zeker niet snel af. Daarbij komt dat verzekeringsproducten snel in de markt gezet moeten worden, wat hoge eisen stelt aan de snelheid waarmee de applicaties aangepast kunnen worden.

Om mee te kunnen bewegen met deze dynamiek heeft Achmea op IT-

gebied meerdere speerpunten geformuleerd. Zo zijn de ontwikkeling van de applicaties en het beheer van IT-infrastructuur samengebracht in één centrale IT-organisatie. Ook is het IT-landschap gestandaardiseerd op een beperkt aantal doelomgevingen, gebaseerd op standaardtechnologieën. Hierbij is gekozen voor SAP als omgeving voor het voeren van polisadministraties in de backoffice. Daarnaast worden Microsoft-producten ingezet om de administraties en andere informatie te ontsluiten op de werkplek. De keuze voor een andere technologie kan alleen gemaakt worden wanneer SAP en Microsoft de gewenste functionaliteit niet kunnen invullen. Door deze standaardisatie wordt zowel het aantal te onderhouden applicaties als de diversiteit in gebruikte technologieën teruggebracht. Achmea wil hiermee een IT-landschap creëren dat effectief en efficiënt aangepast kan worden aan de wensen van de business.

Om deze strategie in de praktijk te brengen, is het noodzakelijk iets te kunnen zeggen over de onderhoudbaarheid van maatwerkapplicaties. Dit helpt bijvoorbeeld bij het bepalen van de meest toekomstvaste applicatie. Daarnaast kan worden geborgd dat applicaties ook in de toekomst onderhoudbaar blijven.

BETROKKENHEID INTERNE AUDITDIENST

Deze ontwikkelingen stellen de interne auditdienst van Achmea voor een uitdaging. De dienst heeft de taak om de *governance*, de risicomangementsystemen en de interne beheersing te beoordelen. Daarnaast wordt de dienst steeds nadrukkelijker een adviseur van de business. Hiermee komt ook het beoordelen van onderhoudbaarheid van applicaties binnen het blikveld van de dienst, en in het verlengde daarvan het adviseren over het verbeteren van de onderhoudbaarheid. Gelukkig zijn er mogelijkheden: onderhoudbaarheid kun je meten en beoordelen.

BEOORDELEN VAN ONDERHOUDBAARHEID

Om onderhoudbaarheid te meten, is het eerst nodig het begrip te definiëren. Een norm voor de beoordeling van software is de ISO 9126-norm voor softwarekwaliteit. Deze norm fungeert vooral als begrippenkader: het legt uit welke aspecten de kwaliteit van software bepalen en uit welke deelaspecten die op hun beurt weer bestaan. De norm is opgenomen in figuur 1.

Een onderdeel van ISO 9126 is het aspect onderhoudbaarheid, dat bestaat uit de deelaspecten Analyseerbaarheid, Veranderbaarheid, Stabiliteit en Testbaarheid. De norm vertelt echter niet *hoe* je die (deel)aspecten kunt meten. ISO 25010, ook bekend als SQuARE (*Systems and software Quality Requirements and Evaluation*).

Ontwikkelhulpmiddelen bieden vaak de mogelijkheid om via een aantal metrieken de onderhoudbaarheid van de broncode te bepalen. Voorbeelden van metrieken zijn: het volume van de broncode en het aantal testprocedures in de broncode. De hulpmiddelen beperken zich echter meestal tot één specifieke programmeertaal, bijvoorbeeld C++ of C#.

Hierdoor is het niet mogelijk om de applicaties die gebaseerd zijn op verschillende programmeertalen onderling te vergelijken. In de praktijk zijn de applicaties echter vaak gebouwd met verschillende programmeertalen. De vraag is daarom op welke manier we onderhoudbaarheid onafhankelijk van de gebruikte programmeertaal kunnen meten, dus op basis van een uniforme set metrieken.

Ontwikkelhulpmiddelen van Microsoft bieden de mogelijkheid om de *maintainability-index* (MI) van de broncode te bepalen. Een nadeel van deze index is dat hij de onderhoudbaarheid in één enkel getal uitdrukt en dat dit getal moeilijk herleidbaar is tot specifieke verbeterpunten in de broncode. Dit is vergelijkbaar met een auditrapport waar de aanbevelingen en audit issues uit zijn verwijderd.

Model van SIG

De Software Improvement Group (SIG) heeft samen met het Duitse TÜViT een model [MODE01] ontwikkeld om de onderhoudbaarheid van applicaties onafhankelijk van de programmeertaal te meten.

Metingen

Stel, we hebben een applicatie waarvan we via metingen de onderhoudbaarheid van de broncode willen ▣



Figuur 1: ISO 9126-norm voor software kwaliteit



bepalen. Deze metingen moeten aan de volgende criteria voldoen om bruikbaar te zijn:

- Programmeertaalafhankelijk, zodat het model ook voor systemen met meerdere programmeertalen gebruikt kan worden.
- Gemakkelijk uitvoerbaar, zodat ze eenvoudig en zonder (grote) investeringen gedaan kunnen worden.
- Begrijpelijk voor ontwikkelaars en uit te leggen aan het management, zodat zij de communicatie tussen de stakeholders kunnen faciliteren.
- Geschikt voor *root-cause* analyse (zoeken naar onderliggende oorzaken van onderhoudbaarheidsproblemen), zodat op basis van de metingen concrete verbetermaatregelen kunnen worden bepaald.

Op basis van deze eisen, uitgebreid onderzoek en praktijkervaring is in de loop der jaren een set meetwaarden samengesteld die de basis vormt van het huidige model:

- *Volume*. De omvang van de code. Hoe meer code, hoe lastiger het is de juiste plek te vinden waar wijzigingen moeten worden doorgevoerd.
- *Duplicatie*. De mate waarin bepaalde stukken code meerdere keren voorkomen in de broncode. Hoe meer duplicatie, hoe lastiger het is om alle plekken te vinden waar gewenste wijzigingen moeten

worden doorgevoerd. Daarnaast is het tijdrovender dezelfde wijziging meerdere keren door te voeren.

- *Unit size*. De lengte van het kleinste mogelijke stuk broncode dat zelfstandig uit te voeren en te testen is. In de programmeertaal C# is dit bijvoorbeeld een *method*. Hoe omvangrijker en complexer de units zijn, hoe lastiger het is wijzigingen door te voeren.
- *Unit complexity*. De mate van complexiteit van een unit. Hoe complexer een unit, hoe lastiger deze te wijzigen is. Er zijn immers meer testcases nodig om alle verschillende paden door een unit af te vangen.
- *Unit interfacing*. Het aantal parameters dat aan een unit moet worden meegegeven om de unit te laten functioneren. Hoe meer dat er zijn, hoe groter de kans is dat een wijziging een onbedoeld neveneffect heeft.
- *Module coupling*. De mate waarin verschillende delen van de code aangeroepen worden door andere. Hoe vaker de code aangeroepen wordt, hoe lastiger deze te wijzigen is zonder onbedoelde neveneffecten te veroorzaken in andere delen van de applicatie.

De set geeft een praktische invulling van het aspect 'onderhoudbaarheid' uit de ISO 9126.

Meetwaarden

Als we onderhoudbaarheid gaan meten, moeten we ook bedenken hoe we deze metingen gaan uitvoeren op een manier die resulteert in betekenisvolle meetwaarden.

Volume

Het aantal regels broncode is een goede basis, die we corrigeren met een productiviteitsfactor om verschillende programmeertalen te kunnen vergelijken.

Duplicatie

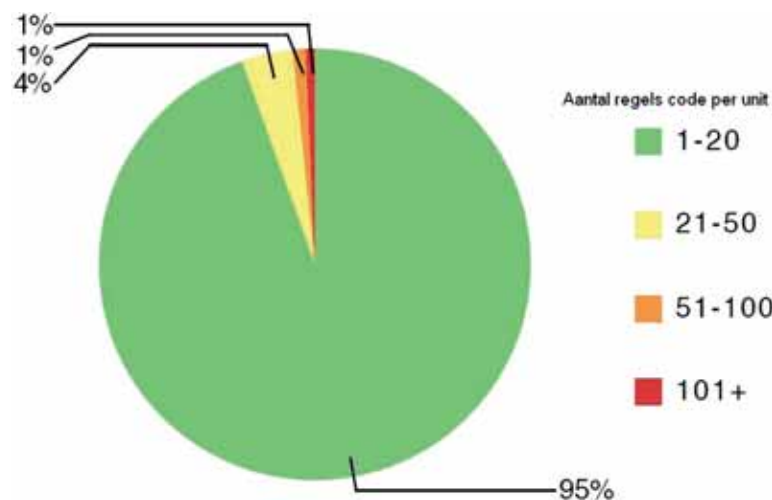
Het percentage regels dat meer dan één keer voorkomt in de code.

Overige meetwaarden

Bij de overige meetwaarden is het ingewikkelder om de stap naar een beoordeling en verbeterpunten te maken. Het blijkt bijvoorbeeld weinig informatief te zijn om de gemiddelde unit size van de totale broncode te berekenen. Ervaring heeft namelijk geleerd dat de gemiddelde unit size van de broncode van applicatie tot applicatie weinig verschilt. De verschillen ontstaan in een beperkt deel van de broncode, en om dit uit te drukken, worden risicocategorieën gehanteerd. Hierbij worden alle regels code ingedeeld in categorieën die afhangen van de lengte van de unit waarin een regel zich bevindt. Zo ontstaat een risicoprofiel voor de unit size van een applicatie. Een voorbeeld is opgenomen in figuur 2. De figuur geeft aan dat slechts 1 procent van de broncode een hoog risico (rood) heeft als het gaat om unit size. Door het aanpassen van deze specifieke 1 procent van de broncode kan de onderhoudbaarheid sterk verbeteren. Ook voor unit complexity, unit interfacing en module coupling kunnen risicoprofielen worden opgesteld.

Benchmarking

We hebben nu meetwaarden voor volume en duplicatie. En we hebben risicoprofielen voor de overige meetwaarden. Het probleem is nu dat



Figuur 2: Voorbeeld risicoprofiel voor unit size

deze meetwaarden op zich nog niet zoveel zeggen. Wat is een aanvaardbaar risicoprofiel voor unit size? Wat is een acceptabel percentage duplicatie? Om hier een antwoord op te geven heeft SIG in de loop der jaren een benchmark met meetwaarden opgebouwd [SCQB01].

Aggregatie meetwaarden tot oordeel

We hebben nu dus zes puntenscores voor de applicatie. Om de metingen te aggregeren tot één enkele score is op basis van expertopinion bepaald welke metingen corresponderen met welke subkarakteristieken van het ISO 9126-model. De uitkomst hiervan is te vinden in figuur 3.

Door de per subkarakteristiek relevante meetwaarden te middelen, komen we tot een sterrenscore per subkarakteristiek. Door die vier scores op hun beurt te middelen, komen we tot één score voor de onderhoudbaarheid van de applicatie, uitgedrukt in een waarde op een 5-puntsschaal (1 tot 5 'sterren'), [GROO12].

DE AUDITAANPAK

IM&IT heeft codeerrichtlijnen opgesteld voor ontwikkelaars om de onderhoudbaarheid van broncode te waarborgen. IM&IT voert echter nog geen structurele metingen uit op het toepassen van deze richtlijnen. Daardoor bestond nog geen inzicht in specifieke knelpunten die een diepgaand onderzoek rechtvaardigen. Om deze reden werd gekozen voor het uitvoeren van een broncode-analyse met SIG. Het doel was om inzicht te verkrijgen in het huidige niveau van onderhoudbaarheid en mogelijke aandachtspunten.

In de audit is de broncode het object van onderzoek. Het is belangrijk alleen die delen van de broncode binnen de scope van de audit op te nemen waar de ontwikkelaar invloed op heeft. In principe valt alle 'handgeschreven' broncode waaruit een

	Volume	Duplicatie	Unit size	Unit complexity	Unit interfacing	Module coupling
Analyseerbaarheid	X	X	X			
Veranderbaarheid		X			X	X
Stabiliteit				X	X	X
Testbaarheid				X	X	

Figuur 3: Weging meetwaarden in subkarakteristieken ISO 9126

applicatie is opgebouwd binnen de scope. Dus ook generieke delen die binnen andere applicaties gebruikt worden. De relevantie hiervan komt later in dit artikel aan bod. Geautomatiseerd gegenereerde broncode, door middel van bijvoorbeeld ontwikkelhulpmiddelen, vallen meestal buiten de scope, omdat de ontwikkelaar geen invloed heeft op de onderhoudbaarheid van die broncode. De broncode van ingekochte (onderdelen van) applicaties kunnen buiten de scope blijven wanneer de ontwikkelafdeling de broncode niet zelf verder ontwikkelt of aanpast voor eigen gebruik. Wanneer de onderhoudbaarheid van ingekochte broncode beoordeeld moet worden, kan dit wellicht beter in een separate audit worden onderzocht omdat de structuur en opbouw van de broncode hoogstwaarschijnlijk afwijken van zelf ontwikkelde applicaties.

Bij Achmea zijn de beoogde gebruiker van het auditrapport en de verantwoordelijke partij voor de audit dezelfde, namelijk de afdeling IM&IT. In veel andere situaties zal dit ook het geval zijn, omdat relatief weinig (externe) assurance gevraagd wordt over de onderhoudbaarheid van broncode.

Het normenkader voor de audit zijn de meetwaarden in het model van SIG. Dit normenkader is gebaseerd op de ISO 9126-norm voor softwarekwaliteit, dat op onderdelen is uitgewerkt door SIG. De begrijpelijkheid van de normen kan een knelpunt zijn voor de verantwoordelijke partij. Of dit het geval is, moet vooraf worden nagegaan en, indien nodig,

dient het management voor aanvang van de audit een toelichting te krijgen over het model en de normen die daaruit volgen.

Object van onderzoek

Achmea Internal Audit en IM&IT hebben gekozen voor het uitvoeren van broncode-analyse van twee applicaties om inzicht te krijgen in het huidige niveau van en mogelijke knelpunten in de onderhoudbaarheid van de broncode. Hiervoor zijn de volgende selectiecriteria opgesteld voor het bepalen van de applicaties binnen de scope van de audit:

- Ze zijn ontwikkeld in een standaardtechnologie binnen een doelomgeving.
- De applicaties zijn recent opgeleverd of er is onderhoud aan uitgevoerd.
- Een spreiding tussen intern en extern ontwikkelde applicaties. Dit maakt het mogelijk verschillen in onderhoudbaarheid tussen extern en intern ontwikkelde applicaties te meten.

Op basis van deze criteria is een tweetal applicaties geselecteerd.

Aanpak broncode-analyse

De geselecteerde applicaties zijn door SIG beoordeeld door middel van een broncode-analyse. Hiervoor kregen zij de beschikking over alle broncode.

De voorlopige bevindingen van de analyse zijn teruggekoppeld naar de ontwikkelaars die onderhoud hebben uitgevoerd aan de applicatie. Dit waren waardevolle sessies, zowel voor SIG om bevindingen scherp te krijgen als voor ontwikkelaars om ▣



achtergrondinformatie bij de bevindingen te krijgen. Het is belangrijk in deze sessies ontwikkelaars in te zetten die de evolutie van de broncode en de applicatie kennen. Zij weten waarom de applicatie op een bepaalde manier is opgebouwd en welke compromissen zijn gesloten gedurende de levensloop van de applicatie. Deze ontwikkelaars kunnen hierdoor de bevindingen snel interpreteren en in een breder perspectief plaatsen.

De auditor heeft in deze fase van de audit een belangrijke verantwoordelijkheid om de bevindingen te wegen. Zoals we later zullen zien, kan het auditoordeel geen gewogen gemiddelde zijn van de scores van de applicaties. Het uiteindelijke oordeel is namelijk mede afhankelijk van de aard en de ernst van de bevindingen. Hierin ligt de toegevoegde waarde van de auditor. Een IT-auditor met ervaring in het auditen van onderhoudbaarheid of het adviseren daarover is belangrijk in deze sessies.

Binnen IM&IT zijn de sessies tussen SIG en de ontwikkelaars positief ontvangen. De belangrijkste redenen hiervoor waren de herkenning van de bevindingen en de uitdaging die de ontwikkelaars kregen om vanuit een helicopterview naar hun eigen expertisegebied, onderhoudbaarheid van de broncode, te kijken.

Een voorbeeld van een mogelijke bevinding is het verschil in omvang tussen modules van een applicatie. Een sterk wisselende omvang van modules duidt vaak op achterstallig onderhoud. Preventief onderhoud kan dan nodig zijn om de structuur van de modules te verbeteren, waardoor in de toekomst wijzigingen efficiënter doorgevoerd kunnen worden in de applicatie. In bepaalde gevallen kan het echter zijn dat onevenwichtigheid een duidelijke reden heeft, zoals het geval kan zijn bij de inkoop van modules. Ontwikkelaars kunnen dit in de sessies toelichten.

Resultaat van de broncode-analyse en oordeel

De analyse van de applicaties en de terugkoppeling naar de ontwikkelaars van IM&IT maakte twee dingen duidelijk:

- De onderhoudbaarheid van beide applicaties was voldoende tot ruim voldoende.
- Beide applicaties lieten verschillen zien in structuur en programmeerstijl. Hoewel dit niet direct iets zegt over de onderhoudbaarheid van de individuele applicaties, maakt dit de applicaties lastiger overdraagbaar tussen ontwikkelteams. Door te standaardiseren op een bepaalde structuur en stijl kunnen verschillende ontwikkel-

teams efficiënter meerdere applicaties onderhouden.

De uitkomsten hebben verder bewezen dat metingen binnen korte tijd een objectief oordeel opleveren over de onderhoudbaarheid en dat ook snel specifieke verbeterpunten geformuleerd kunnen worden.

De scores voor onderhoudbaarheid van de individuele applicaties zijn uitgedrukt op een schaal van 1 t/m 5. De scores vormen het belangrijkste uitgangspunt voor het bepalen van het auditoordeel. Het is belangrijk bij het bepalen ook de ernst van de bevindingen mee te nemen, een mathematische formule om vanuit de scores het oordeel te bepalen is niet te geven.

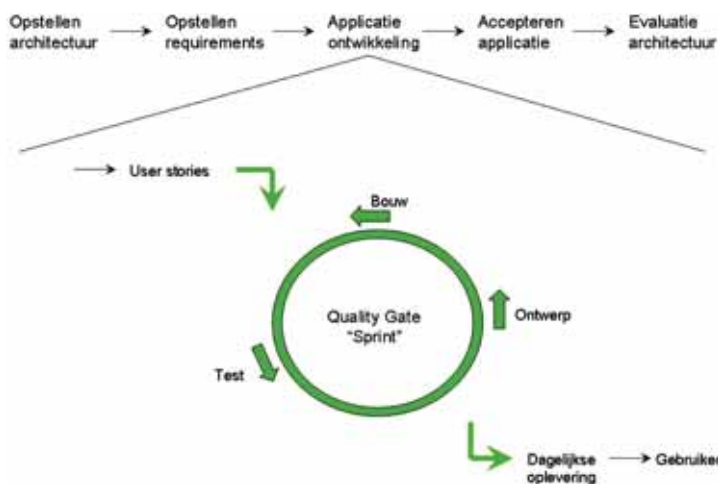
ADVISEREN OVER ONDERHOUDBAARHEID IN ONTWIKKELPROCES

Zoals eerder vermeld, heeft Achmea ervoor gekozen zich te concentreren op een aantal doelomgevingen, gebaseerd op standaardtechnologieën (SAP en Microsoft-producten).

IM&IT ontwikkelt applicaties vooral in SharePoint-technologie en hanteert hierbij het Microsoft Solutions Framework. In dit Framework is een *rapid-application-development* (RAD) methodiek geïntegreerd. Een kernpunt van RAD is de nauwe samenwerking tussen ontwikkelaars en businessanalisten waarbij in korte, dagelijkse ontwikkelcycli (zogenaamde *sprints*) nieuwe applicatiefunctionaliteit wordt opgeleverd. Toekomstige gebruikers krijgen hierdoor snel een beeld van de functionaliteit en kunnen direct bijsturen in de ontwikkeling.

IM&IT heeft voor de besturing van software-ontwikkelprojecten een proces gedefinieerd (zie figuur 4).

De figuur geeft aan dat het uitgangspunt voor een ontwikkelcyclus de *User Stories* zijn, deze geven de gewenste functionaliteit weer. In de



Figuur 4: Software ontwikkelproces Achmea IM&IT

sprints wordt vervolgens deze functionaliteit ontwikkeld.

Voor de beheersing van het ontwikkelproces is een aantal zogenaamde *Quality Gates* (QG) gedefinieerd. Dit zijn momenten in de tijd waarop specifieke kwaliteitskenmerken, waaronder onderhoudbaarheid, gemeten worden. De onderhoudbaarheid van de broncode wordt dagelijks gemeten na afronding van iedere sprint. In het proces past de borging van de kwaliteit van de broncode dan ook bij afronding van de sprint: dat is het moment om knelpunten te signaleren en ook om direct in te kunnen grijpen als de kwaliteit van de code op één van de aspecten onder het vereiste niveau daalt. Tijdig ingrijpen zorgt voor het snel oplossen van knelpunten. Zo blijft de onderhoudbaarheid gewaarborgd. Wanneer bijvoorbeeld het aantal regels broncode per unit size te sterk toeneemt, kan hier direct op ingespeeld worden door te analyseren wat de achterliggende oorzaak van deze grote units is. Als dit bijvoorbeeld voortkomt uit het ontwerp van de software, dan kan dit worden aangepast en op basis daarvan de broncode worden verbeterd. Wanneer deze aanpassingen pas plaatsvinden nadat de volledige applicatie is opgeleverd, betekent dat niet alleen een grotere inspanning. Het doorvoeren van aanpassingen is dan ook aanzienlijk complexer omdat de kans groot is dat aanpassingen gevolgen hebben voor meerdere delen van de broncode.

IM&IT meet na iedere sprint dagelijks de kwaliteitscriteria zoals genoemd in figuur 5.

Hiermee definieert IM&IT een minimumniveau voor deze criteria, echter nog zonder daaruit specifieke verbeterpunten in de broncode te kunnen definiëren. Op dit punt komt de audit naar onderhoudbaarheid goed van pas. Het doel is de maintainability-index in de QG te vervangen [ONTW01] door de meer specifieke criteria en risicoanalyses zoals gehan-

Quality Gate : Sprint	
1.	Code Coverage voor framework is minimaal 80%.
2.	Code Coverage voor Business Logic/Access Layer is minimaal 60%.
3.	Code Coverage voor User Interface is minimaal 40%.
4.	Alle bugs met severity <i>critical</i> en <i>important</i> hebben de status <i>closed</i> .
5.	Aantal Static Code Analysis meldingen is 0. Suppressed warnings zijn beargumenteerd.
6.	Aantal unit tests dat niet succesvol is, is 0.
7.	De code is gecompileerd met 0 errors en 0 warnings. Suppressed warnings zijn beargumenteerd.
8.	De maintainability index op assembly niveau is minimaal 70.
9.	De maintainability index op methode niveau is minimaal 40.
10.	Het sprint release is gepackaged zodat het automatisch gedeployed kan worden.

Figuur 5: Inhoud Quality Gate 6

teerd in het SIG/TÜViT-model. Dit ondersteunt en stimuleert ontwikkelaars in het verbeteren van de onderhoudbaarheid van applicaties.

TOT SLOT

IT-auditors kunnen met de aanpak in dit artikel hun blik richten op het waarborgen en verbeteren van de onderhoudbaarheid van maatwerkapplicaties. Zij kunnen hierbij zowel een oordeel geven over onderhoudbaarheid van specifieke applicaties, als adviseren over het waarborgen van onderhoudbaarheid in een ontwikkelproces. ■

Literatuur

[GRO012] Groot, J. de en Visser, J., De waarde van softwareproducten bepalen, *de IT-Auditor*, 2011, nr. 1, blz. 24-31.

[MODE01] Heitlager, I., Kuipers, T., Visser, dr. ir. J., A Practical Model for Measuring Maintainability, *Proceedings of the 6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007)*, pages 30-39, IEEE Computer Society Press, 2007 (http://www.sig.eu/blobs/Research/Scientific%20publication/HeitlagerKuipersVisser-Quatic2007_08.pdf).
[SCQB01] Baggen, R., Schill, K., Visser, J., Standardized Code Quality Benchmarking for Improving Software Maintainability, *Proceedings of the 4th International Workshop on Software Quality and Maintainability (SQM 2010)*, March 15, 2010, Madrid, Spain, 2010 (<http://www.sig.eu/blobs/Research/Scientific%20publication/SQM2010-StandardizedCodeQuality.pdf>).
[ONTW01] Visser, J., *SIG/TÜViT Evaluation Criteria Trusted Product Maintainability, with guidance for producers*, SIG, 2011 (<http://www.sig.eu/blobs/Services%20Certificering/SIG-TUViT%20Evaluation%20Criteria%20Trusted%20Product%20Maintainability.pdf>)
[NORE01] NOREA, RE-GIDS, 2011, 6-C1 Raamwerk Assurance-opdrachten, artikel 23.



ing. M. (Mark) van der Krift RE is sinds 2006 werkzaam als IT-auditor bij de Interne Audit Dienst van Achmea. Hij heeft meerdere jaren ervaring als software ontwikkelaar en richt zich op audits in Microsoft omgevingen.



M. (Marijn) Dessens is senior consultant bij SIG. Hij heeft ruime ervaring in het beoordelen van onderhoudbaarheid van software bij financiële instellingen.