



Een kwaliteitsmodel voor maatwerksoftwaresystemen ter ondersteuning van portfoliomanagement

Om een maatwerksysteem goed te kunnen beheren en daarbij efficiënt met de beschikbare middelen om te kunnen gaan, is kennis van de verschillende (meest technische) aspecten van het maatwerksysteem noodzakelijk. Om binnen een samenwerkingsverband waarin verschillende maatwerksystemen actief zijn sturingsinformatie over de betrokken systemen te verzamelen, is een kwaliteitsmodel ontwikkeld dat inzicht geeft in de staat van deze maatwerksystemen. Als de informatie uit dit kwaliteitsmodel beschikbaar is, kunnen betere beslissingen worden genomen en kan de continuïteit van de functionaliteit die deze systemen bieden, worden gewaarborgd.

Dit artikel is tot stand gekomen naar aanleiding van werkzaamheden die KPMG tijdens een aantal adviesopdrachten voor overheidsorganisaties heeft uitgevoerd. Het sluit aan op en is een nadere uitwerking van het 'Toetskader voor beheer en onderhoud' dat het Adviescollege ICT-Toetsing in september 2025 publiceerde ([AICT25]).



Kevin Bankersen MSc
is business consultant bij
Schuberg Philis.



Drs. J.M.A. Koedijk
CISA CISM
is partner bij KPMG
Advisory.

INLEIDING

Er zijn organisaties en samenwerkingsverbanden, zeker binnen de rijksoverheid, waar een portfolio van maatwerksoftwaresystemen (inclusief RPA [Robotic Process Automation]- en AI-toepassingen) moet worden onderhouden, beheerd en vernieuwd. Gezien de aard van de ondersteunde processen is maatwerksoftware veelal noodzakelijk. Die processen zijn immers specifiek vormgegeven om invulling te geven aan de kenmerken en behoeften van een gemeente, een regio of het land.

Maatwerksoftware betekent echter ook dat de organisatie er zelf voor verantwoordelijk is om de systemen bij de tijd te houden en tegelijkertijd, onder meer door application lifecycle management (ALM), de continuïteit van de dienstverlening langdurig te borgen. Dit vraagt om een planmatige aanpak die meerdere jaren beslaat. Een aanpak ook waarin gelijkmatig wordt omgegaan met de hulpbronnen, waaronder de medewerkers, externe leveranciers en financiën. Een dergelijk plan dient, in onze optiek, te worden opgesteld en uitgevoerd onder leiding van het management van de organisatie.

Een verzameling systemen vraagt natuurlijk om portfoliomanagement. Daarvoor zijn ook al verschillende methoden beschreven. Veelal richten deze aanpakken zich met name op optimalisatie van de aansluiting bij de businessprocessen en de met de instandhouding van het systeem samenhangende kosten ([Groo15a], [Groo15b]). In een verzameling systemen die ieder een eigen onafhankelijke functie hebben, biedt een dergelijke methode nog onvoldoende houvast voor de planvorming. Wat in een dergelijke situatie nodig is, is kennis van de technische, functionele en juridische aansluiting van het systeem bij de toekomst.

Als er de afgelopen jaren iets blijkt uit de vele ICT-mislukkingen en de rapporten van het Adviescollege ICT-Toetsing, dan is het wel dat kennis van de staat van de softwaresystemen bij de leiding van de organisaties veelal ontbreekt of witte vlekken vertoont. Waar moet je ook op letten? De onderhoudbaarheid van de broncode uitgedrukt in sterren? De leeftijd van de software of de hardware? De oplostijd van incidenten of sowieso het aantal incidenten per week? De aansluiting bij andere systemen in de 'nabijheid'? Er zijn veel verschillende variabelen die een rol kunnen spelen. Wat echt nodig is, is een overzicht dat met enige diepgang inzicht geeft hoe het ICT-systeem er in de breedte voor staat.

Voor een samenwerkingsverband binnen de overheid waarin meerdere overheden en uitvoeringsorganisaties betrokken zijn, heeft KPMG rond deze vragen een model ontwikkeld. Het model beoogt in een rapportage van enkele pagina's per softwaresysteem de staat inzichtelijk

te maken. Het idee is dat met dit model, samengevoegd met inzicht in de functionele ontwikkeling van het systeem en de kosten die met noodzakelijke wijzigingen en de instandhouding van het systeem gemoeid zijn, de leiding van het samenwerkingsverband goede beslissingen kan nemen over hoe de beschikbare middelen over het applicatieportfolio kunnen worden verdeeld.

We bespreken in dit artikel het model dat we in samenwerking met de bestuursondersteuning van het samenwerkingsverband hebben opgesteld. Ook geven we ruimte aan de reacties van verschillende stakeholders van de diverse systemen. Ten slotte geven we aan hoe een organisatie of samenwerkingsverband een eigen model rond het applicatieportfolio kan opstellen, invullen en bijhouden.

AANLEIDING

Rond een aantal voorzieningen die een cruciale rol spelen in de dienstverlening aan burgers, ontstond een samenwerkingsverband van overheidspartijen. De kern van deze systemen is dat de functionaliteit nog voor jaren beschikbaar zal moeten zijn. Voorheen was elke organisatie, elk organisatieonderdeel en/of elke afdeling (hierna 'organisatie') afzonderlijk verantwoordelijk voor het beheer, onderhoud en lifecycle management van het door de organisatie verzorgde systeem. In het samenwerkingsverband werd beoogd om op centraal niveau beslissingen te nemen over verdeling van middelen in de verwachting dat dit efficiënter kan zijn en leidt tot een meer gelijkmatig gebruik van de middelen.

Dat bleek zo eenvoudig nog niet. Veel van de voorgestelde activiteiten werden door de beheerorganisaties als strikt noodzakelijk voorgesteld. Vanuit het perspectief van ontstane achterstanden zou dat nog enigszins verklaarbaar kunnen zijn. Maar toen bleek dat de claim op middelen altijd groter was dan het beschikbare budget en het leek alsof achterstanden niet werden ingelopen (of steeds weer vanuit een andere hoek opkwamen), ontstond de behoefte aan een meer onafhankelijk oordeel over de staat van de diverse softwaresystemen. Zonder een dergelijk objectief beeld bleek daadwerkelijke op 'inhoud' gebaseerde sturing op de besteding van middelen vanuit de leiding van het samenwerkingsverband niet mogelijk.

HET ONTSTAAN VAN HET KWALITEITSMODEL

Bij KPMG worden veel onderzoeken naar technische kwaliteit van softwaresystemen uitgevoerd. Het valt daarbij op dat heel vaak kerninformatie over systemen niet direct voorhanden is. Vragen als 'Welke documen-

tatie is beschikbaar?’, ‘Hoeveel regels broncode omvat de software bij benadering?’, ‘Welke technologieën worden gebruikt?’, ‘Zijn er automatische testen?’ en ‘Is er technische schuld?’ kunnen veelal niet direct worden beantwoord. Dat is bijzonder. Het is immers deze informatie die de basis dient te vormen voor de sturing op de technologische ontwikkeling. Niet zelden leiden deze onderzoeken er dan ook toe dat de prioriteiten in de ontwikkeling en het beheer van de onderzochte systemen veranderen.

Op basis van onze ervaringen is een aantal organisaties op weg geholpen met kwaliteitsdoelen voor de maatwerksoftware die ze in beheer hebben. Hiervoor werden eenvoudig te bepalen metrieken gebruikt, zoals methodecomplexiteit, codeduplicatie en unit-testdekking, waarbij ook de kwaliteit van ondersteunende componenten (‘dependencies’) werd meegenomen. Er wordt hierbij bewust gekozen voor kwaliteitsdoelen in plaats van een ‘harde’ normering omdat hier voor een organisatie wat betreft prioriteit en ambities keuzes te maken zijn (zie ook [Amor13] en [Koed19]). De ervaring bij deze organisaties toonde aan dat periodiek rapporteren (afhankelijk van de veranderingsbehoefte één tot vier keer per jaar) en daarbij de verandering (verbetering/verslechtering) meenemen, medewerkers motiveert om daadwerkelijk de kwaliteit te verbeteren en de kwaliteitsdoelen te benaderen.

Met kwaliteitsdoelen voor de software is er een begin van een model. Duidelijk is echter ook dat het gewenste kwaliteitsmodel aanvulling behoeft op veel andere aspecten die niet de maatwerksoftware zelf betreffen om binnen het portfolio goed sturing te kunnen geven. Denk hierbij aan de hardware, architectuur en de inrichting van de beheerorganisatie. Om een stap verder te komen richting een volledig model zijn er diverse inspiratiesessies en interviews gehouden met betrokkenen, zowel vanuit de leiding als vanuit de beheerders van individuele systemen, binnen het samenwerkingsverband. Vanuit deze gesprekken is meer een omvattend kwaliteitsmodel opgesteld. Dit model is daarna verder aangescherpt in een aantal reviewsessies met de opdrachtgever, die daarbij ook de reacties van beheerorganisaties meenam.

KWALITEITSMODEL VOOR SOFTWARESYSTEMEN

Na een grondige inventarisatie, in meerdere iteraties, van details die bij besluitvorming van belang kunnen zijn, zijn die voor het kwaliteitsmodel samengevat in acht aspecten waarover moet worden gerapporteerd. Voor elk van deze aspecten zijn vragen opgesteld die het beeld van het betreffende aspect bepalen. Het is daarbij de bedoeling dat die vragen relatief eenvoudig en met weinig inspanning te beantwoorden zijn vanuit de beheerorga-

Het valt op dat heel vaak kerninformatie over systemen niet direct voorhanden is

nisatie zodat dit zo min mogelijk impact heeft op andere noodzakelijke werkzaamheden. Wel zullen verschillende afdelingen of organisatieonderdelen een bijdrage moeten leveren; vragen over privacy en informatiebeveiliging vereisen immers een andere expertise dan die over technische schuld.

De acht aspecten zijn (in willekeurige volgorde):

- *Softwarearchitectuur*
Een goede beschrijving van hoe het systeemtechnisch in elkaar zit is belangrijk omdat een dergelijke beschrijving inzicht geeft in wat een systeem wel en niet kan. Daarbij kan ook worden aangegeven hoe verschillende non-functional requirements (NFR's) zoals performance en schaalbaarheid worden behaald. Sowieso is een duidelijke beschrijving van de non-functional requirements waarmee rekening is gehouden belangrijk.
- *Technologie*
Technologie zegt iets over de ouderdom van het systeem. Hoewel oude Cobol-software op zich niet slechter functioneert dan systemen in het thans populaire Python, zegt het wel iets over de aantrekkingskracht van een dergelijke ontwikkelomgeving op (toekomstige) softwareontwikkelaars (die het softwarebeheer en -onderhoud moeten uitvoeren) en de 'investeringszin' van technologieproviders. Vergelijkbare verhalen zijn ook voor onder meer hardware, netwerktechnologie en ontwikkelomgevingen van toepassing. Deze informatie geeft een indicatie op welke termijn vervanging van een component of van het hele systeem kan worden verwacht – hoewel in de praktijk vaak blijkt dat met lapmiddelen verlenging van de levensduur mogelijk is.
- *Technische schuld*
Technische schuld zijn bekende achterstanden in het systeemonderhoud waarvoor nog werk moet worden verricht om het systeem weer up-to-date te krijgen ([Amor13]). Er wordt gesproken van 'schuld' omdat er veelal sprake is van achterstallig onderhoud door onvoldoende aandacht (lees: geld) voor het beheer dat noodzakelijk is om de kwaliteit van de software op peil te houden, in lijn met de eerdergenoemde vastgestelde kwaliteitsdoelen.
- *Open-sourcestrategie*
Zeker binnen de overheid wordt het gebruik van open-sourcecomponenten belangrijk gevonden om de afhankelijkheid van grote ICT-bedrijven te verminderen. Een strategie voor het gebruik van deze componenten, die ook beschrijft of de door de organisatie ontwikkelde software ook weer ter beschikking wordt gesteld aan een community, is daarbij nodig. Uiteraard mag een lijst met gebruikte open-sourcecomponenten inclusief versie nummers ook hier niet ontbreken.

- *Wet- en regelgeving*
Wet- en regelgeving is in het overheidsdomein vaak een belangrijke bron van de wijzigingsbehoefte. Ook komt het voor dat de laatste wijzigingen in wet- en regelgeving nog niet in de systeemfunctionaliteit zijn verwerkt. En dat betreft niet alleen de wijziging in het operationele domein, zoals sociale zekerheid, maar zeker ook AI-, privacy-, informatiebeveiligings- en archiefwetten. Een belangrijke wet die vaak een rol speelt bij de inhuur van leveranciers, is de Aanbestedingswet, waaraan uiteraard ook moet worden voldaan.
- *Privacy & informatiebeveiliging*
Het beschermen van persoonsgegevens is van groot belang, juist omdat de overheid gerechtigd is om zeer privacygevoelige gegevens, bijvoorbeeld adressen van vips of giften aan politieke partijen of kerkgenootschappen, te verwerken. Omdat deze gegevens niet in verkeerde handen mogen vallen, is aandacht voor privacy en informatiebeveiliging van groot belang.
- *Beheer en releasemanagement*
De eenvoud van beheer en de beheersbaarheid van het systeem inclusief de omgeving zijn, hoewel niet eenvoudig in één metriek te vangen, belangrijke kenmerken. Veel systemen voldoen nog niet aan de vereisten voor 'continuous integration' (CI) en 'continuous deployment' (CD) en beschikken niet over een goed geautomatiseerde DevOps-omgeving. Kennis over testbaarheid en de snelheid waarmee een release naar de productieomgeving kan worden gebracht, is echter belangrijk om het systeem goed te kunnen beoordelen.
- *Risicomanagement*
Geen enkel systeem functioneert zonder risico's. Het is daarom belangrijk dat beheerorganisaties de risico's kennen, maatregelen nemen om risico's te voorkomen en/of de impact te verkleinen en zich voorbereiden op het mogelijk (toch) optreden van risico's.

In figuur 1 zijn de acht aspecten met hun belangrijkste vragen weergegeven.

Veel van de vragen hebben niet een kwantitatief antwoord, maar vragen eerder om een korte kwalitatieve beschrijving van de situatie. Dit heeft mogelijk wel tot gevolg dat de antwoorden binnen een samenwerkingsverband met eerste metingen enigszins naar elkaar toe zullen moeten convergeren. Desalniettemin geeft in onze overtuiging beantwoording van de vragen, tezamen met een korte beschrijving van doel en context van het systeem (zoals aantal gebruikers en belanghebbenden, gevolgen van het missen van een deadline of een verstoring), een goede basis voor besluitvorming over prioriteiten en de verdeling van middelen.

Privacy & informatiebeveiliging

- Wordt er voldaan aan de securityvoorwaarden?
- Worden er testen gedaan om te kijken of de security actueel is?
- Hoe wordt de security in de toekomst gewaarborgd?
- Welke externe componenten worden er gebruikt (met versienummer) en bevatten deze bekende kwetsbaarheden?

Wet- & regelgeving

- Wordt er voldaan aan wet- en regelgeving?
- Hoe wordt er op toekomstige wet- en regelgeving geanticipeerd?
- Moet er structureel veel aangepast worden bij nieuwe wet- en regelgeving?

Risicomanagement

- Worden risico's gemonitord en gerapporteerd?
- Wat is het huidige inzicht en beeld wat betreft die risico's?
- Is Quality Assurance ingericht?

Open source

- Is er een open-sourcestrategie?
- Welke open-sourcecomponenten worden er gebruikt (met versienummers)?
- Welke open-sourcestandaarden worden gebruikt?
- Zijn toekomstige onderdelen open source?

Beheer & releasemanagement

- Is de voorziening gemakkelijk te beheren en wat wordt er voor beheerders gedaan zodat het beheersbaar blijft?
- Hoe vaak wordt er gereleased en wat als de release niet goed gaat?
- Zijn er onderscheidende OTAP-omgevingen?
- Hoe is de verhouding in capaciteit tussen de omgevingen en zijn er realistische (geanonimiseerde) testdata beschikbaar?
- Kunnen er performancetesten worden uitgevoerd?
- Kan met alle benodigde/gekoppelde systemen worden getest?

Softwarearchitectuur

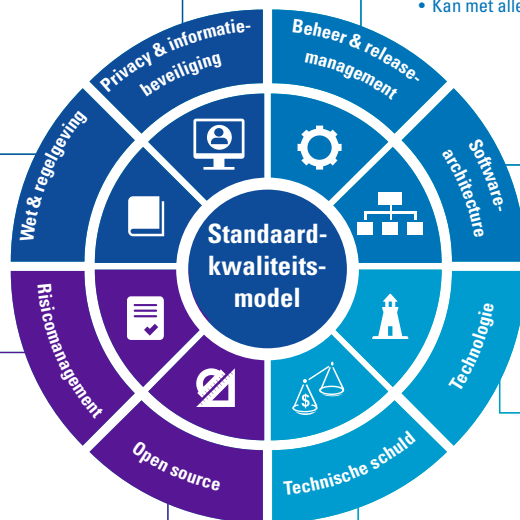
- Hoe is de architectuur beschreven? (Volgt de beschrijving de NORA?)
- Volgt de beschrijving de moderne standaarden?
- Hoe voorziet de architectuur in schaalbaarheid en modulariteit?
- Zijn er non-functional requirements (NFR's) voor onder meer performance, security en autorisatie vastgelegd?

Technologie

- Uit welke ontwikkelomgeving is de software opgebouwd?
- Welke standaardcomponenten worden er gebruikt, inclusief technologieversies (besturingssysteem, database, middleware, etc.), en worden die nog door de leveranciers ondersteund?
- Hoe is het hostingplatform opgebouwd (virtual server, cloud, on-premise, etc.)?
- Met welke externe systemen en architectuurpatronen worden er gegevens uitgewisseld?

Technische schuld

- Hoe groot is de technische schuld?
- Welke maatregelen worden er genomen om de technische schuld op te lossen?
- Hoe wordt technische schuld in de toekomst voorkomen/gemini-maliseerd?



Figuur 1. De acht aspecten van het kwaliteitsmodel voor softwaresystemen.

IDE = Integrated Development Environment (zoals Visual Studio of IntelliJ)

OTAP = Ontwikkel-, Test-, Acceptatie- en Productieomgeving

NORA = Nederlandse OverheidsReferentie Architectuur

NFR = non-functional requirements

Quality Assurance zijn de tweedelijns maatregelen om de kwaliteit van het product te borgen.

ONTVANGST

Na afstemming met de bestuursondersteuning is voor verdere afstemming contact gezocht met een aantal beheerorganisaties van de systemen. Zij waren minder enthousiast om het model meteen te gaan gebruiken. Een van de eerste dingen die zij opmerkten is dat het verzamelen van de informatie (kostbare) tijd/capaciteit zou kosten. Verder werd er gewezen op het feit dat de verschillende voorzieningen zeer verschillend waren en daardoor in hun ogen niet vergelijkbaar waren.

Een bijzondere reactie kregen we van een beheerorganisatie over de vraag naar een gedetailleerde lijst van de gebruikte componenten. Volgens hen was het bestuur in deze mate van detail niet geïnteresseerd. Inderdaad is wellicht een vertaalslag op deze informatie nodig. Maar een lijst van standaardcomponenten die gebruikt worden, inclusief versienummers, is iets wat voor elk maat-

werksysteem nodig is. De consternatie na het gevonden informatiebeveiligingsprobleem in de log4j-component in december 2021 ([Hofm21]), waarbij het voor veel organisaties lang onduidelijk bleef of ze aan deze kwetsbaarheid blootgesteld waren, heeft aangetoond dat een goede inventarisatie van componenten snel inzicht geeft als een veelgebruikte component een probleem heeft. Elke beheerorganisatie van een maatwerksysteem die 'in control' wil zijn, heeft als het goed is die lijst al klaarliggen.

Het belangrijkste bezwaar echter, zo gaven de beheerorganisaties toe, is dat ze de noodzaak voor centrale informatieverstrekking en aansturing niet zagen. Vanuit het perspectief dat de beheerorganisaties al verantwoordelijk waren voor hun systeem voordat het samenwerkingsverband was ontstaan, is dat wellicht te begrijpen. Het heeft echter tot gevolg dat aansturing vanuit het samenwerkingsverband niet goed mogelijk blijft en dat dus prioriteitsstelling, efficiencyvoordelen en lifecycle

management over de individuele systemen heen niet mogelijk zijn.

Uiteindelijk is het bewijs van een kwaliteitsmodel natuurlijk vooral gelegen in het gebruik ervan. We zijn in gesprek met een beheerorganisatie die mee wil werken aan het in kaart brengen van de gedefinieerde kwaliteitsaspecten van hun eigen maatwerksoftwaresysteem. Wij hopen het kwaliteitsmodel dan succesvol te kunnen beproeven zodat de beheerorganisatie beslissingen, die de toekomst van de systemen beïnvloeden, kan nemen waarbij kennis van de systemen een rol speelt.

CONCLUSIE

Om weg te komen van incidenten en onvermijdelijkheden is het noodzakelijk om maatwerksoftwaresystemen goed te onderhouden en application lifecycle management uit te voeren. Dit vereist dat er vanuit het bestuur een plan is waarin beschikbare middelen evenwichtig en met aandacht voor de hoogste prioriteiten worden verdeeld. Het is niet mogelijk goed sturing te geven aan het onderhoud van systemen zonder inhoudelijke kennis van die systemen. In een samenwerkingsverband is dat niet anders. Het is dan ook onze overtuiging dat een kwaliteitsmodel voor maatwerkssystemen dat de leiding inzicht verschaft, op termijn onvermijdelijk is. Uit de reacties van de geconsulteerde beheerorganisaties leiden wij af dat het in dit artikel gepresenteerde kwaliteitsmodel van voldoende kwaliteit is om een eerste ronde van informatie over systemen op te halen. Wat ons betreft kan daar niet snel genoeg mee gestart worden. Het eerdergenoemde 'Toetskader voor beheer en onderhoud' ([AICT25]) zien wij daarbij als een extra aanbeveling. Als gevolg zullen er betere beslissingen kunnen worden genomen en kan ondertussen ook, met de opgedane ervaring, het kwaliteitsmodel verder worden verbeterd.

Literatuur

- [AICT25] Agentschap ICT-toetsing (2025, 29 september). *Toetskader voor beheer en onderhoud*. Geraadpleegd op: <https://www.adviescollegeicttoetsing.nl/onze-werkwijze/toetskader-beheer-en-onderhoud/toetskader-beheer-en-onderhoud>
- [Amor13] Amoraal, J.M., Lanzani, G., Kuiters, P., & Koedijk, J.M.A. (2013). Grip op de kwaliteit van software. *Compact* 2013/2. Geraadpleegd op: <https://www.compact.nl/articles/grip-op-de-kwaliteit-van-software-2/>
- [Groo15a] Groosman, J.H.L. & Hoss, J.M. (2015). Application Portfolio Optimization Publication: Getting Ready for Digital Transformation. *Compact* 2015/2. Geraadpleegd op: <https://www.compact.nl/articles/application-portfolio-optimization>
- [Groo15b] Groosman, J.H.L., Kuiters, P., & Knip, S. (2015). Removing Application Entropy. *Compact* 2015/2. Geraadpleegd op: <https://www.compact.nl/articles/removing-application-entropy>
- [Hofm21] Hofmans, T. (2021, 14 december). Wat weten we nu over Log4Shell, de kwetsbaarheid in de log4j-library? *Tweakers*. Geraadpleegd op: <https://tweakers.net/reviews/9614/wat-weten-we-nu-over-log4shell-de-kwetsbaarheid-in-de-log4j-library.html>
- [Koed19] Koedijk, J.M.A. & Knottnerus, J.M. (2019). Zet je niet af, maar stel een doel. *Compact* 2015/2. Geraadpleegd op: <https://www.compact.nl/articles/zet-je-niet-af-maar-stel-een-doel-2/>

Over de auteurs

Kevin Bankersen MSc is *Compact*-redacteur en senior manager bij KPMG. Sinds 2016 heeft hij aan uiteenlopende projecten binnen de software-engineering gewerkt. Zo is hij betrokken bij het opzetten en beoordelen van softwarearchitecturen, het ontwikkelen van technische ontwerpen, het onderzoeken van softwarekwaliteit (broncode en architectuur), het uitvoeren van technische due-diligence- en integratieonderzoeken en het opzetten en integreren van DevOps-toolinginfrastructuur. Hij is onlangs als business consultant in dienst getreden bij Schuberg Philis.

Drs. J.M.A. Koedijk CISA CISM trad in 1997 toe tot de joint venture van KPMG met Cambridge Technology Group als teamleider softwareontwikkeling. Sindsdien ondersteunen hij en zijn team klanten bij het ontwerpen en ontwikkelen van software en geven zij technisch advies over softwarearchitectuur en ontwikkelingsmogelijkheden. Door middel van performancetesten en codebeoordelingen biedt het team ook inzicht in softwarekwaliteitskwesaties, wat resulteert in aanbevelingen voor het ontwikkelingsproces en softwareverbetering. Als partner en grondlegger van de Software Quality-praktijk van KPMG Advisory in Nederland staat Joost zij aan zij met zijn klanten om de uitdagingen van software-engineering aan te gaan.