

# **Kwaliteitskenmerken van softwareproducten: specificeren, evalueren en certificeren**

**J.J.M. Trienekens en E.P.W.M. van Veenendaal**

*Kwaliteit van softwareproducten is reeds lange tijd een ongrijpbaar concept, zowel voor gebruikers van softwareproducten als voor systeemontwikkelaars. Voor gebruikers is met name het specificeren van kwaliteitskenmerken een terrein vol vraagtekens. Voor ontwikkelaars is veelal niet duidelijk welke kwaliteitskenmerken van een softwareproduct noodzakelijk zijn en op welke wijze deze kunnen worden ingebouwd in een softwareproduct. Nog moeilijker is de taak van beoordelaars, c.q. testers, namelijk het vaststellen of de noodzakelijke en/of de gewenste kwaliteit in een softwareproduct aanwezig is.*

*Dit hoofdstuk beschrijft oplossingsrichtingen voor de genoemde vraagstukken. Het bepalen van de kwaliteitskenmerken van een softwareproduct begint met de identificatie van de kwaliteitsbehoeften van gebruikers. Deze behoeften komen altijd voort uit de karakteristieken van een bedrijfssituatie of bedrijfssysteem. Een bedrijfssysteem kan worden gekarakteriseerd op basis van drie soorten karakteristieken:*

- bedrijfsproces karakteristieken*
- gebruikers karakteristieken*
- software product karakteristieken.*

*Deze karakteristieken vormen een basis voor de specificatie van de kwaliteitskenmerken van een softwareproduct. Zo'n specificatie wordt verder geconcretiseerd aan de hand van metrieken. Vervolgens kunnen ontwikkelaars de specificatie gebruiken om kwaliteit in te bouwen in een softwareproduct. Ook dient een kwaliteitsspecificatie als referentiekader voor het opzetten van testplannen en voor het evalueren en certificeren van softwareproducten.*

## **Algemeen**

### ***Softwarekwaliteit: aspecten en stromingen***

De jaren negentig worden door vele deskundigen gezien als een decennium waarin de aandacht voor kwaliteit van softwareproducten sterk is toegenomen. In bedrijfsleven en maatschappij zijn gebruikers niet alleen meer vertrouwd geraakt met de toepassingsmogelijkheden van softwareproducten, ook de wensen en eisen ten aanzien van kwaliteit hebben zich sterk ontwikkeld. Gebruikers zijn niet langer tevreden met softwareproducten die "enkel" aan de gestelde functionele specificaties voldoen en tijdig worden opgeleverd. Softwareproducten dienen tevens te voldoen aan een heel scala van zogeheten kwaliteitseisen. Daarmee is kwaliteit minstens zo belangrijk geworden als op tijd leveren en tegen afgesproken kosten.

Echter, het specificeren en kwantificeren van de kwaliteit van softwareproducten is geen sinecure. We geven twee redenen daarvoor. Enerzijds is het moeilijk voor gebruikers om kwaliteitsbehoeften expliciet te maken. Uitspraken zoals: "het systeem moet prettig zijn in het gebruik", en "het systeem dient te kunnen meegroeien met de organisatie" zijn vaak te vaag of te algemeen van aard. Anderzijds ondervinden ontwikkelaars veel problemen bij het nemen van

de juiste maatregelen om kwaliteit “in te bouwen” in een softwareproduct. Voorbeelden daarvan zijn: het op gestructureerde wijze ontwikkelen van softwareproducten (o.a. GOTO-statements vermijden), het testen van systemen met geavanceerde tools, het modulair opzetten van systemen, etc.

Hoewel op allerlei fronten door ontwikkelaars wordt gewerkt aan kwaliteitsverbetering is vaak onduidelijk op welke manier deze inspanningen werkelijk bijdragen aan de kwaliteit zoals die door gebruikers wordt gewenst of wordt vereist. Twee belangrijke benaderingen kunnen op het terrein van kwaliteitsverbetering worden onderscheiden, respectievelijk de procesgerichte benadering en de productgerichte benadering. In het navolgende worden beide benaderingen kort beschreven.

### ***Procesbenadering***

Deze benadering richt zich met name op het formaliseren, structureren en verbeteren van de ontwikkelprocessen. De gebruikers van softwareproducten zijn nauwelijks betrokken bij de procesbenadering. Het is moeilijk vast te stellen wat de uiteindelijke effecten zijn van procesverbeteracties voor de gebruikers of de bedrijfsprocessen waarbinnen de softwareproducten worden toegepast. Onduidelijk blijft vaak of de juiste procesverbeteringen worden doorgevoerd en wat die verbeteringen bijdragen aan de kwaliteit van een softwareproduct voor een gebruiker. Enkele voorbeelden van procesbenaderingen zijn ISO 9000-3, het Capability Maturity Model en SPICE (Software Process Improvement and Capability Determination). Ondanks de vele inspanningen op dit terrein van kwaliteitsverbetering blijven vragen onbeantwoord zoals:

- wat is het nut van procesverbeteracties voor gebruikers?
- hoe kunnen gebruikerswensen en -eisen ten aanzien van productkwaliteit een rol spelen bij het inrichten van verbeteracties van software-ontwikkelaars?
- hoe kunnen ontwikkelaars productkwaliteit uitleggen in voor gebruikers begrijpelijke termen?

### ***Productbenadering***

Deze benadering richt zich op de specificatie en evaluatie van softwarekwaliteitskenmerken. Gesteld wordt dat kwaliteitskenmerken expliciet kunnen worden gespecificeerd en kunnen worden ingebouwd in een softwareproduct. Productkwaliteit kan vervolgens worden geevalueerd, bijvoorbeeld door middel van het testen van een softwareproduct.

Voor het specificeren van softwarekwaliteitskenmerken kan tegenwoordig een eenduidig begrippenkader en een consistente terminologie worden gehanteerd. De ISO9126 standaard biedt een modelmatige onderbouwing en definities van kwaliteitskenmerken. Voorbeelden van kwaliteitskenmerken zijn respectievelijk: betrouwbaarheid, gebruiksvriendelijkheid, efficiency (o.a. tijdgedrag) en onderhoudbaarheid. In diverse onderzoekprojecten is in de afgelopen jaren gestreefd naar meetbaarheid van deze productkenmerken. Voorbeelden zijn het QUINT-project en het ESPRIT-project SCOPE. Afwisselend staan in deze projecten onderwerpen centraal zoals de specificatie van relaties tussen kwaliteits- en produktkenmerken, de selectie van metrieken, en de ontwikkeling van evaluatieplannen. Een van de meer recentere ontwikkelingen is de identificatie en specificatie van kwaliteitsbehoeften van gebruikers. Op basis daarvan kunnen de kwaliteitskenmerken van een softwareproduct worden gespecificeerd.

Dit hoofdstuk richt zich in het navolgende op de productbenadering. Eerst komt het specificeren aan de orde en vervolgens het evalueren en testen van softwareproductkwaliteit.

## **Productkwaliteit specificeren**

Deze paragraaf gaat eerst in op de definities van kwaliteitskenmerken. Vervolgens komen aan de orde het belang van het stellen van prioriteiten ten aanzien van kwaliteitskenmerken en het proces van het specificeren van kwaliteitskenmerken.

### ***Kwaliteitskenmerken***

Reeds in de jaren zeventig werd beschreven hoe softwarekwaliteit kan worden onderverdeeld in kwaliteitskenmerken. In de decennia daarna is veel onderzoek verricht naar het beschrijven van kwaliteit aan de hand van kenmerken, subkenmerken en metrieken.

In de sterk toegenomen stroom publikaties over de softwareproductkwaliteit lijkt momenteel overeenstemming te zijn bereikt over de gedefinieerde en gekwantificeerde kwaliteitskenmerken. De *International Organization for Standardization* (ISO) en de *International Electrotechnical Commission* (IEC) heeft een aantal "standaard" kwaliteitskenmerken gedefinieerd. Deze standaard is een belangrijke stap op weg naar consensus over productkwaliteit tussen makers en afnemers van softwareproducten. De standaard bevat definities van zes hoofdkenmerken van productkwaliteit en stelt een verdere onderverdeling van kwaliteit voor in een aantal subkenmerken:

- functionality (functionaliteit), bestaande uit vijf subkenmerken: suitability, accuracy, compliance, interoperability en security;
- reliability (betrouwbaarheid), onderverdeeld in de subkenmerken maturity, fault tolerance, recoverability en availability;
- usability (bruikbaarheid), met de subkenmerken understandability, learnability, operability en likability;
- efficiency (efficiëntie), onder te verdelen in time behaviour en resource behaviour;
- maintainability (onderhoudbaarheid), bestaande uit vier subkenmerken: analysability, changeability, stability en testability;
- portability (portabiliteit), bestaande uit vijf subkenmerken: adaptability, installability, co-existence, conformance en replaceability.

Het vertrouwen van een gebruiker in een softwareproduct zal toenemen naarmate het evalueren en beoordelen van de kwaliteitskenmerken grondiger gebeurt. In de praktijk blijkt het testen van softwareproducten een steeds belangrijkere discipline te worden. Voordat in dit hoofdstuk nader wordt ingegaan op het testen, en met name het testen van de kwaliteitskenmerken van een softwareproduct, wordt eerst aandacht besteed aan het identificeren van kwaliteitsbehoeften en het specificeren van kwaliteitskenmerken.

### ***Kwaliteitsprofiel: prioriteiten voor kwaliteitskenmerken***

In de praktijk is het onmogelijk om alle kwaliteitskenmerken van een softwareproduct met een zelfde diepgang te testen. Gezien de tijd en de inspanningen die nodig zijn voor diepgaand testen dienen prioriteiten te worden gesteld. Dit hoeft geen probleem te zijn mits die prioriteiten op inzichtelijke wijze en goed onderbouwd tot stand komen. Uitgangspunt voor het bepalen van prioriteiten zijn de kwaliteitsbehoeften van gebruikers. Met name dient een softwareproduct op die kenmerken te worden getest die belangrijk worden gevonden door gebruikers, bijvoorbeeld op grond van de karakteristieken van bedrijfsprocessen waarbinnen het softwareproduct ondersteuning dient te bieden. We geven twee voorbeelden. Van een tekstverwerkingsapplicatie kan het belangrijker zijn om het "gebruiksgemak" meer diepgaand te testen dan de "betrouwbaarheid", wat uiteraard niet betekent dat "betrouwbaarheid" van een

tekstverwerker geheel onbelangrijk is. Voor een softwareproduct dat in een nucleaire installatie wordt gebruikt geldt vaak het omgekeerde: “betrouwbaarheid” wordt vele malen belangrijker geacht dan “gebruiksgemak”. Echter, ook hier geldt dat dit niet betekent dat “gebruiksgemak” totaal irrelevant is. Dit soort relaties tussen bedrijfssituaties en softwarekwaliteitskenmerken zijn recentelijk nader onderzocht en uitgewerkt.

Ook zijn richtlijnen opgesteld om te kunnen aangeven hoe diepgaand het evalueren van kwaliteitskenmerken van een softwareproduct, afhankelijk van de situatie, dient plaats te vinden. In deze richtlijnen wordt een viertal *evaluatie-niveaus* voorgeschreven, met een oplopende mate van diepgang (van niveau D tot en met niveau A). Het vereiste evaluatieniveau voor een specifieke bedrijfssituatie wordt bepaald aan de hand van factoren zoals economische risico's in geval van bedrijfsprocesuitval, continuiteitsrisico's voor een organisatie, afdeling of taakgebied en risico's met betrekking tot veiligheid.

Een beschrijving van geprioriteerde productkwaliteitskenmerken en hun evaluatieniveaus wordt een kwaliteitsprofiel genoemd. Een kwaliteitsprofiel formaliseert en concretiseert het begrip productkwaliteit voor zowel ontwikkelaars als gebruikers. Figuur 1 toont een vereenvoudigd voorbeeld (beperkt tot hoofdkwaliteitskenmerken) van een kwaliteitsprofiel.

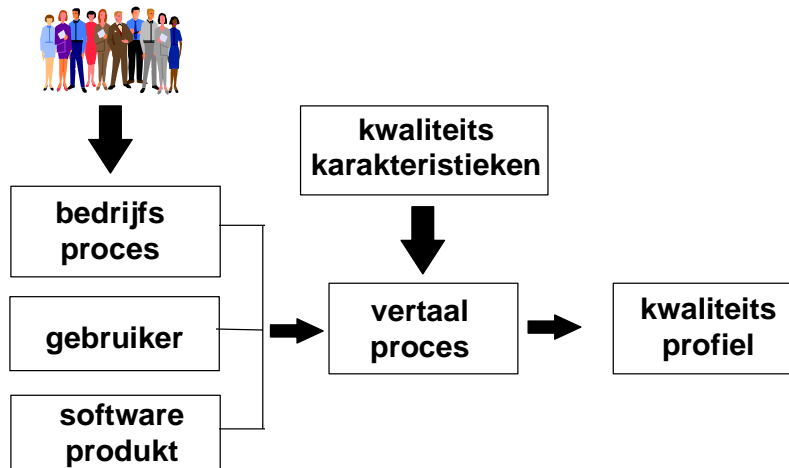
	Evaluatieniveau				
	-	D	C	B	A
Functionality				X	
Reliability			X		
Usability		X			
Efficiency		X			
Maintainability			X		
Portability	X				

Figuur 1: Voorbeeld van een kwaliteitsprofiel

In het navolgende wordt nader ingegaan op een methode voor het ontwikkelen van een kwaliteitsprofiel.

### ***Het specificeren van software product kwaliteit***

Wensen en behoeften van gebruikers, op basis van het specifieke bedrijfssysteem of taakgebied waarbinnen zij werken, dienen het uitgangspunt te vormen voor het specificeren van een kwaliteitsprofiel van een softwareproduct. Bedrijfssystemen kunnen worden beschreven aan de hand van drie soorten karakteristieken: respectievelijk karakteristieken van bedrijfsprocessen, karakteristieken van gebruikers en karakteristieken van het software product zelf. In Figuur 2 is het specificatieproces modelmatig weergegeven:



Figuur 2: Specificatiemodel voor productkwaliteit

Drie stappen worden onderscheiden. In de eerste stap wordt gebruik gemaakt van gestructureerde vragenlijsten om de karakteristieken van het bedrijfssysteem te kunnen vaststellen. In de tweede stap worden de systeemkarakteristieken gerelateerd aan kwaliteitskenmerken. In de derde stap komt de prioriteitstelling ten aanzien van softwarekwaliteitskenmerken tot stand. In deze paragraaf wordt met name ingegaan op de bepaling van de karakteristieken van een bedrijfssysteem, c.q. bedrijfsproces-, gebruikers- en softwareproductkarakteristieken.

#### ***Karakteristieken van een bedrijfsproces.***

Op basis van principes uit de systeemtheorie, met name de cybernetica, worden de volgende karakteristieken onderscheiden:

- het *aantal* bedrijfsprocesvariabelen en hun *onderlinge afhankelijkheid*
- de *beheersbaarheid* en *voorspelbaarheid* van bedrijfsprocesvariabelen
- de *gevoeligheid* en *stabiliteit* van een bedrijfsproces
- het *reactiepatroon* van een bedrijfsproces.

We geven twee voorbeelden van verbanden tussen bedrijfsproceskarakteristieken en kwaliteitskenmerken:

- naarmate het aantal bedrijfsprocesvariabelen en hun onderlinge afhankelijkheid binnen een bedrijfsproces groter is, is een bedrijfsproces moeilijker te beheersen; de behoefte aan kwaliteit voor wat betreft “usability”, met name de subkenmerken “understandability” en “operability”, is dan groter.
- naarmate een bedrijfsproces gevoeliger is voor interne of externe veranderingen, bijvoorbeeld veranderingen in verantwoordelijkheden, taakpakketten, communicatie, etc. is de behoefte groter om aanpassingen te kunnen verrichten aan het softwareproduct dat dat bedrijfsproces ondersteunt. In termen van kwaliteitskenmerken betekent dit dat de behoefte aan “maintainability”, met name het subkenmerk “changeability” groter is.

#### ***Gebruikers karakteristieken***

Gebruikers zijn direct dan wel indirect betrokken bij de selectie, het gebruik of beheer van een softwareproduct. Menselijke factoren dienen dan ook een rol te spelen bij het bepalen en specificeren van kwaliteitskenmerken. Voorbeelden van gebruikerskarakteristieken zijn

respectievelijk de *ervaring* die gebruikers hebben met vergelijkbare softwareproducten, met softwareproducten in het algemeen en met de bedrijfsprocessen waarbinnen zij werken, het *opleidingsniveau* van de gebruikers, de *omvang* van de gebruikersgroep.

We geven twee voorbeelden van relaties tussen gebruikerskarakteristieken en kwaliteitskenmerken:

- naarmate een softwareproduct is bedoeld voor grotere aantallen onervaren gebruikers met relatief lage opleidingsniveau's is de behoefte groter aan "understandability" en "learnability" (beide zijn subkenmerken van "usability").
- naarmate meer invoer-georiënteerde eindgebruikers dienen te werken met een softwareproduct dient het "gebruiksgemak" hoger te zijn. Dit stelt hogere eisen aan het "usability"-subkenmerk "operability" en het "efficiency"-subkenmerk "time-behaviour".

### ***Karakteristieken van de technische omgeving***

Karakteristieken van de technische omgeving: onderscheid wordt gemaakt tussen drie dimensies: het type software systeem, het soort gebruik en de technische infrastructuur:

- Het *type* software systeem: voorbeelden van typen systemen zijn gegevensverwerkende systemen, beslissingsondersteunende systemen, kennissystemen. Verder zijn indelingen mogelijk zoals standaard pakketten versus maatwerk systemen voor specifieke afnemers.
- Het soort *gebruik*: bijvoorbeeld het aantal uren per dag dat een softwareproduct operationeel is, of het onderscheid tussen "on-line" en "batch" georiënteerde systemen.
- De bestaande *infrastructuur*: hardware platform, client-server, netwerk, etc.

We geven twee voorbeelden van relaties tussen karakteristieken van de technische omgeving en softwarekwaliteitskenmerken:

- naarmate bij een gegevensverwerking systeem een groter aantal batch-jobs dienen te worden verricht is de "recoverability" (een subkenmerk van "reliability") belangrijker.
- naarmate een softwareproduct meer eigenschappen heeft van een standaardpakket is "portability" een belangrijker kwaliteitskenmerk.

## **Testen van productkwaliteit**

### ***State-of-the-practice***

Wanneer de stand van zaken ten aanzien van testen in de praktijk wordt bestudeerd, dan valt op dat in de afgelopen jaren een groot aantal verbeteringen is doorgevoerd in het testproces. Veel organisaties maken tegenwoordig gebruik van het zogenaamde V-model, vaak in combinatie met een gedetailleerd faseringsmodel voor het 'black-box'testen. Een speciale testomgeving is redelijk standaard en er zijn testtechnieken beschikbaar voor allerlei testdoelen en toepassingen. Computer Aided Software Testing (CAST) is bovendien door de jaren heen sterk verbeterd, waardoor testers nu over een aantal belangrijke hulpmiddelen kunnen beschikken. De belangrijkste ontwikkeling is wellicht dat organisaties tegenwoordig veelal bereid zijn tijd en geld te investeren in testactiviteiten en dat testen eindelijk wordt beschouwd als een professionele activiteit.

Er dient te worden opgemerkt dat het merendeel van de doorgevoerde verbeteringen gericht zijn op het testproces zelf (doing the things right) in plaats van op de effectiviteit ervan (doing the right things). Bovendien zijn de meeste verbeteringen vrij technisch van aard en zijn ze

nauwelijks gericht op de gebruiker. Belangrijke vragen die in dit verband dienen te worden gesteld zijn:

- leidt het verbeterde testproces ook tot meer inzicht bij de gebruikers omtrent de mate waarin het softwareproduct in hun behoeften voorziet?
- Worden vanuit het oogpunt van de gebruiker de juiste tests uitgevoerd, met andere woorden wordt er getest ten aanzien van ‘fitness=for-use’?

De belangrijkste activiteit in het testproces als moet worden vastgesteld *wat* moet worden getest (doing the right things), is het bepalen van de teststrategie. De teststrategie geeft aan wat getest gaat worden en met welke diepgang getest gaat worden.

### *Teststrategie*

Het bepalen van de teststrategie vindt plaats tijdens de planningsfase van het testproces. Tijdens deze fase wordt de basis gelegd voor een beheersbaar en kwalitatief hoogstaand testproces. Het bepalen van de teststrategie is voor het testmanagement een middel om te communiceren met de opdrachtgever, gebruikers, ontwikkelaars en andere betrokkenen over de organisatie en strategische keuzes die ten aanzien van het testproces moeten worden gemaakt. Bij het bepalen van de teststrategie dient expliciet te worden vastgesteld wat getest gaat worden en met welke diepgang getest gaat worden. Er dienen keuzes te worden gemaakt aangezien het onmogelijk is om een softwareproduct volledig te testen. In theorie is het wellicht mogelijk alle functionaliteiten en kwaliteitskenmerken voor 100% af te dekken, maar geen enkele organisatie heeft daarvoor genoeg tijd en geld beschikbaar. Bij het bepalen van de teststrategie wordt getracht de belangrijkste functionaliteiten en kwaliteitskenmerken van een softwareproduct te identificeren. Het doel hiervan is de best haalbare dekking ten aanzien van de juiste functionaliteiten en de juiste kwaliteitskenmerken van het softwareproduct te bewerkstelligen.

Binnen het proces van strategiebepaling kunnen een aantal stappen worden onderscheiden waarin kwaliteitskenmerken een expliciete rol spelen:

- *Selecteren kwaliteitskenmerken*  
In overleg met alle betrokkenen wordt een aantal kwaliteitskenmerken geselecteerd, op basis van bijv. ISO 9126, die relevant zijn voor het onderhavige softwareproduct. In de ideale situatie zijn de relevante kwaliteitskenmerken reeds vastgelegd in de functionele specificatie. Ter ondersteuning van het selectieproces kan een risicotaxatie worden uitgevoerd, zodat het mogelijk is de belangrijkste kwaliteitskenmerken meer gefundeerd en in detail te bepalen. Het selectieproces kan worden omschreven als informeel en is vrij technisch georiënteerd vanwege de aard van de meeste kwaliteitskenmerken. Gebruikers weten veelal niet wat er precies wordt bedoeld met kenmerken, zoals portabiliteit, onderhoudbaarheid of betrouwbaarheid.
- *Bepalen relatief belang*  
Het relatieve belang van de geselecteerde kwaliteitskenmerken moet worden bepaald, uitgedrukt in een percentage. Het gaat niet om het berekenen van exacte percentages, omdat dat in dit stadium niet zinvol en onmogelijk is. Het doel van deze stap is om in samenspraak met de opdrachtgever en andere betrokkenen te komen tot een algemeen beeld van het relatieve belang van de geselecteerde kenmerken. Door percentages toe te kennen aan elk van de geselecteerde kenmerken dienen de belanghebbenden een bepaalde mate van consensus te bereiken over het relatieve belang.
- *Vaststellen acceptatiecriteria*  
Er moet worden vastgesteld aan welke criteria het te testen softwareproduct moet voldoen

opdat het kan worden geaccepteerd. De acceptatiecriteria worden vastgesteld per geselecteerd kwaliteitskarakteristiek. Uiteraard speelt het (relatief) belang van een kwaliteitskarakteristiek een belangrijke rol bij het vaststellen van het acceptatiecriterium.

Het resultaat van de eerste twee stappen wordt vaak weergegeven in de zogenaamde strategiematrix (zie tabel hierna). De tweedimensionale matrix geeft aan welke kwaliteitskenmerken relevant zijn en hoe groot hun relatieve belang is. Het softwareproduct in het navolgende voorbeeld bestaat uit een drietal deelsystemen (S1, S2 en S3). De relaties tussen de kwaliteitskenmerken en de drie deelsystemen zijn beschreven in de strategiematrix. Een 'x' betekent dat het betreffende kwaliteitskarakteristiek van toepassing is op het betreffende deelsysteem, een 'xx' betekent dat de combinatie kwaliteitskarakteristiek/deelsysteem speciale aandacht vergt, hetgeen dient te worden vastgesteld in meer diepgaande tests.

	S1	S2	S3	Relatief belang
Functionaliteit	xx	x	xx	40%
Betrouwbaarheid	xx	x	xx	20%
Bruikbaarheid	x			10%
Efficiëntie	x			10%
Onderhoudbaarheid	x	x	x	20%
Portabiliteit				-
				100%

Figuur 3: Strategiematrix

Hoewel het proces voor het bepalen van een teststrategie redelijk gestructureerd is en reeds in een groot aantal projecten in de praktijk met succes is toegepast, heeft het nog een aantal belangrijke tekortkomingen. Het proces is erg informeel van aard en kan niet worden gekenmerkt als herhaalbaar en reproduceerbaar. Mede hierdoor is de kwaliteit van het resultaat c.q. de teststrategie sterk afhankelijk van de kennis en de kunde van het betrokken testmanagement. De voornaamste tekortkoming is echter gelegen in het feit dat het communicatiemiddel de (technisch georiënteerde) kwaliteitskenmerken zijn en de gebruikersbehoeften ten aanzien van kwaliteit geen expliciete rol spelen bij het bepalen van de teststrategie. Het proces van strategiebepaling is derhalve niet erg gebruikersvriendelijk; het heeft geen expliciete gebruikers- en/of bedrijfsoriëntatie. Zoals reeds aangegeven weten gebruikers veelal niet wat exact wordt bedoeld met de diverse kwaliteitskenmerken. Als tevens, hetgeen vaak het geval is, geen gebruik wordt van standaards inzak de definitie van een kwaliteitskenmerk, zoals bijv. ISO 9126, dan neem de (spraak)verwarring met gebruikers nog verder toe. Desalniettemin wordt geprobeerd tijdens het testen de "fitness-for-use" vast te stellen.

#### *Kwaliteitsprofiel ook voor testers*

Het eerder beschreven kwaliteitsprofiel komt tot stand op basis van de kwaliteitsbehoeften van de gebruikers. Het kan derhalve worden gekenschetst als een gebruikersgeoriënteerd kwaliteitsprofiel. Een vergelijking van het kwaliteitsprofiel met de strategiematrix, levert een groot aantal overeenkomsten. Beiden geven aan welke kwaliteitskenmerken belangrijk zijn in een specifieke (bedrijfs)situatie en in beide komt het belang van de geselecteerde kwaliteitskenmerken tot uiting. Bij het kwaliteitsprofiel wordt dit bereikt door middel van de



evaluatie-niveaus en bij de strategiematrix door middel van het relatieve belang. In het proces van het bepalen van de relevante kwaliteitskenmerken ligt bij beide methoden de nadruk op een intensieve communicatie met de gebruikers. Het grootste verschil tussen de twee methoden is gelegen in het feit dat het kwaliteitsprofiel wanneer het toegepast wordt met behulp van een evaluatiemodel, daadwerkelijk gebruikersgeoriënteerd is. Een ander verschil is het ontbreken van een expliciete risicoanalyse binnen het evaluatiemodel, hoewel er wel een diepgaand contextonderzoek plaatsvindt op basis van de kenmerken van de bedrijfssituatie. Dit betekent dat indien het kwaliteitsprofiel wordt gebruikt als basis voor het testen van een softwareproduct, een aantal (kleine) aanpassingen noodzakelijk is op basis van een uitgevoerde risicoanalyse en wellicht als gevolg van beperkingen in de beschikbare tijd, ter bepaling van de “definitieve” teststrategie.

Samenvattend kan worden geconcludeerd dat een kwaliteitsprofiel een basis vormt voor een gebruikersgeoriënteerde teststrategie. Dit resulteert vervolgens in een meer gebruikersgeoriënteerd testproces. Door middel van de uitgevoerde test wordt het mogelijk ene uitspraak te doen over de “fitness-for-use” van het softwareproduct.

### **Conclusie**

Een kwaliteitsprofiel dient uitgangspunt te zijn bij zowel het specificeren en evalueren, c.q. testen van productkwaliteit. Het kan worden toegepast binnen diverse fasen van het systeemontwikkelingsproces. Het proces ter bepaling van een kwaliteitsprofiel op basis van het evaluatiemodel wordt in de ideale situatie uitgevoerd door ontwikkelaars, testers en gebruikers tezamen. Diverse projecten hebben inmiddels de waarde van het kwaliteitsprofiel aangetoond.