

TESTEN OP BASIS VAN DE KWALITEITSBEHOEFTE VAN GEBRUIKERS

(Drs. E.P.W.M. van Veenendaal CISA en Dr. ir. J.J.M. Trienekens)

De kwaliteit van een softwareprodukt is voor zowel gebruikers als testers veelal een ongrijpbaar begrip. Ondanks interessante resultaten tijdens onderzoek en in de praktijk, kunnen gebruikers de kwaliteitskarakteristieken van door hen gewenste softwareproducten niet in voldoende mate beschrijven. Als gevolg hiervan weten testers veelal niet welke kwaliteitskarakteristieken gewenst en noodzakelijk zijn, hetgeen ertoe leidt dat het bijna onmogelijk is om softwareproducten te testen vanuit gebruikersoptiek.

Het bepalen van de gewenste kwaliteitskarakteristieken begint met communiceren met de gebruiker over zijn of haar kwaliteitsbehoeften. Dit dient te resulteren in een beschrijving van de bedrijfssituatie waarin de gebruiker het softwareprodukt wil toepassen. Een beschrijving van de bedrijfssituatie dient te bestaan uit de drie volgende onderdelen:

- de bedrijfsproces karakteristieken
- de gebruikers karakteristieken
- softwareprodukt karakteristieken.

Vanuit de beschrijving van de bedrijfssituatie kunnen kwaliteitskarakteristieken worden afgeleid op een semi-deterministische wijze. Kwaliteitskarakteristieken kunnen worden opgesplitst in subkarakteristieken, die vervolgens kunnen gerelateerd aan indicatoren en metrieken. Deze karakteristieken, bepaald op basis van de kwaliteitsbehoeften van gebruikers, bieden vervolgens een gedegen basis voor een gebruikers georiënteerd testproces.

De beschreven methode is initieel ontwikkeld door de TU-Eindhoven en wordt momenteel verder uitgewerkt, en gevalideerd bij KEMA Nederland bv in het corporate research project Software Product Quality en in de Europese ESPRIT projecten SPACE-UFO en MultiSpace.

GBRUIKERS WILLEN KWALITEIT

Vele deskundigen op het gebied van software ontwikkeling hebben de jaren negentig gekenschetst als het kwaliteitstijdperk [2]. Het toenemende gebruik van informatiesystemen in de maatschappij heeft de gebruikers ervan vertrouwd gemaakt met de mogelijkheden. Niet langer is men tevreden met softwareproducten die "enkel" aan de gestelde functionele specificaties voldoen, op tijd en tegen aanvaardbare kosten worden opgeleverd. Gebruikers van softwareproducten willen tevens dat de opgeleverde systemen voldoen aan hun kwaliteitseisen. In de praktijk blijkt dat gebruikers veelvuldig de kwaliteit van hun softwareproducten onvoldoende vinden in relatie tot hun behoeften, hoewel hen op overtuigende wijze - zij het niet in hun eigen taal - is uitgelegd dat het systeem aan allerlei (technische) normen voldoet: het produkt is getest met de meest geavanceerde tools, de complexiteit is relatief gering en er is geen gebruik gemaakt van GOTO-statements. Deze eigenschappen zijn voor de gebruiker echter onzichtbaar tijdens het gebruik van de softwareproducten. De technische eigenschappen en kenmerken van het systeem dienen te zijn gebaseerd op de behoeften en wensen van de gebruiker; slechts dan kunnen gebruikers het vertrouwen hebben dat het produkt voldoet aan hun kwaliteitseisen en is een bevredigend en comfortabel gebruik ervan gewaarborgd. De software industrie tracht in deze behoeften te voorzien. Momenteel kunnen twee belangrijke stromingen worden onderscheiden in het kader van het streven naar kwaliteitsverbetering: de procesgerichte benadering en de produktgerichte benadering.

Procesbenadering

De procesgerichte benadering kan worden beschouwd als een indirecte benadering voor het verbeteren van de kwaliteit van softwareproducten. Deze benadering richt zich met name op het identificeren, structureren en verbeteren van de ontwikkelprocessen. De gebruikers van de softwareproducten zijn nauwelijks betrokken bij de procesbenadering. Voorbeelden van de procesgerichte benadering zijn ISO 9000-3 [12], TickIT [22], het Capability Maturity Model [9] en het SPICE-project (Software Process Improvement and Capability

Determination [7]). Ondanks de goede resultaten die met deze benadering zijn behaald, blijven er vragen over, zoals:

- Hoe kan kwaliteit door de ontwikkelaars expliciet en concreet worden gemaakt?
- Hoe kunnen gebruikers worden overtuigd worden van de kwaliteit van de softwareproducten?
- Wie bepaalt of beloften en verwachtingen ten aanzien van softwarekwaliteit realistisch zijn?

Produktbenadering

Om deze vragen te kunnen beantwoorden is een produktgerichte benadering van kwaliteit vereist. Vaak wordt de produktbenadering als een tegenpool van de procesbenadering beschouwd, maar in feite zijn de beide benaderingen complementair. De produktbenadering zorgt voor metingen op basis waarvan de ontwikkelprocessen op een meer efficiënte en effectieve wijze kunnen worden verbeterd. De produktbenadering is gebaseerd op de gedachte dat kwaliteit van een softwareprodukt kan worden bereikt door enerzijds de relevante kwaliteitskarakteristieken van het softwareprodukt te identificeren en te specificeren en anderzijds de gerealiseerde kwaliteitseigenschappen te beoordelen c.q. te evalueren. Voorbeelden van kwaliteitskarakteristieken van softwareproducten zijn (volgens ISO 9126) betrouwbaarheid, bruikbaarheid, efficiency en onderhoudbaarheid. Voorbeelden van onderzoeksprojecten op dit gebied zijn het QUINT-project [21, 26] op nationaal niveau, en de ontwikkeling van de standaard ISO-9126 [13] en diverse Europese ESPRIT-projecten zoals METKIT, SCOPE en SQUID op internationaal niveau. In deze projecten worden met name de kwaliteitskarakteristieken verder uitgewerkt tot produktkenmerken en wordt gewerkt aan de totstandkoming van metrieken. Hoewel een bepaalde mate van objectiviteit en kwantificering is bereikt, wordt noch in het QUINT-project, noch in de genoemde Europese ESPRIT-projecten op gestructureerde wijze aandacht besteed aan het bepalen van de gebruikersbehoeften en de eisen die vanuit bedrijfssituatie waarin een softwareprodukt gebruikt zal worden, worden gesteld. Dit heeft tot gevolg in dat het beoordelen van de kwaliteit van het softwareprodukt vanuit het oogpunt van de gebruiker vrijwel onmogelijk is. In recent aangevangen ESPRIT-projecten zoals SPACE-UFO (User Focus) en MultiSpace, gericht op de kwaliteit van multimedia toepassingen, wordt getracht de technische benadering van softwarekwaliteit en de gebruikers georiënteerde benadering op basis van kwaliteitsbehoeften tot elkaar te brengen.

Bij de behandeling van de methode staat het testen van softwareproducten centraal. Testen is immers de meest gehanteerde werkwijze om softwareproducten te beoordelen. Allereerst wordt de huidige stand van zaken met betrekking tot testen en het bepalen van de teststrategie beschreven. Vervolgens wordt op basis van een aantal kwaliteitsconcepten een nieuwe benadering voor het testen van softwarekwaliteit geïntroduceerd, waarbij het expliciet identificeren en specificeren van de kwaliteitsbehoeften van gebruikers en kwaliteitseisen vanuit de bedrijfssituatie als uitgangspunt wordt gehanteerd.

TESTEN VAN SOFTWAREPRODUCTEN

De praktijk

In de software industrie bestaat nog geen eenduidige en algemeen geldende definitie van het begrip "testen". De vele soorten tests met uiteenlopende doelen, de vage begrenzingen en de veelal informele toepassing van het fenomeen maken het opstellen van een eenduidige definitie lastig. In de praktijk wordt onder testen vaak eenvoudigweg verstaan meten, vergelijken, uitproberen enzovoort. Deze praktijkgerichte benadering is echter ontoereikend in de huidige professionele bedrijfssituaties. De standaard ISO 8402 [11] geeft de volgende definitie van testen: "activiteiten zoals meten, onderzoeken, beproeven, keuren met kalibers van één of meer kenmerken van een produkt of dienst en het vergelijken van de uitkomsten met de gestelde eisen, om te bepalen of aan deze voorwaarde is voldaan." Bij testen gaat het om het ontdekken van fouten: testen toont het gebrek aan kwaliteit aan. Een bruikbare definitie van het begrip testen, op basis van TMap, luidt: "testen is een proces van plannen, voorbereiden en meten met als doel de kenmerken van een softwareprodukt vast te stellen en het verschil tussen de actuele en de vereiste status aan te tonen [20]."

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 soorten testdoelen en toepassingen. CAST (Computer Aided Software Testing) is

bovendien door de jaren heen sterk verbeterd, waardoor testers nu over een aantal bruikbare 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. De uitspraken over de verbeteringen van het testproces worden ondersteund door de resultaten van een onderzoek dat door de NGGO is uitgevoerd ten aanzien van het testen van informatiesystemen.

Het testproces (bron: inventariserend rapport NGGO [17])	
Gemiddelde testinspanning als percentage van de totale projectinspanning	20%
Hanteert richtlijnen/voorschriften voor het uitvoeren van de testactiviteiten	49%
Neemt de uit te voeren tests in de planning van iedere afzonderlijke fase op	75%
Neemt de testgegevens op in het eindproduct van de fase	52%
Onderkent het nut van testvoorbereiding	91%
Voert tests uit	96%
Onderkent de noodzaak van hertesten	84%

Er dient echter 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 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, m.a.w. 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 de 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 is onmogelijk om een softwareproduct volledig te testen; in theorie is het wellicht mogelijk alle functionaliteiten en kwaliteitskarakteristieken 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 kwaliteitskarakteristieken van een softwareproduct te identificeren. Het doel hiervan is de best haalbare dekking ten aanzien van de juiste functionaliteiten en de juiste kwaliteitskarakteristieken van het softwareproduct te bewerkstelligen.

Binnen het proces van strategiebepaling [20] kunnen een aantal stappen worden onderscheiden waarin kwaliteitskarakteristieken een expliciete rol spelen:

- *Selecteren kwaliteitskarakteristieken*

In overleg met alle betrokkenen wordt een aantal kwaliteitskarakteristieken geselecteerd, op basis van bijv. ISO 9126, die relevant zijn voor het onderhavige softwareproduct. In de ideale situatie zijn de relevante kwaliteitskarakteristieken reeds vastgelegd in de functionele specificatie. Ter ondersteuning van het selectieproces kan een risico-taxatie worden uitgevoerd zodat het mogelijk is de belangrijkste kwaliteitskarakteristieken 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

kwaliteitskarakteristieken. Gebruikers weten veelal niet wat er precies bedoeld wordt met karakteristieken zoals portabiliteit, onderhoudbaarheid of betrouwbaarheid.

- *Bepalen relatief belang*
Het relatieve belang van de geselecteerde kwaliteitskarakteristieken 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 karakteristieken. Door percentages toe te kennen aan elk van de kwaliteitskarakteristieken dienen de belanghebbende 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 geselecteerde kwaliteitskarakteristiek. Uiteraard speelt het (relatief) belang van een kwaliteitskarakteristiek een belangrijke rol bij het vaststellen van het acceptatiecriterium.
- *Bepalen testtechnieken*
Tijdens deze stap dient te worden bepaald op welke wijze c.q. met welke testtechnieken, de kwaliteit van het softwareproduct zal worden vastgesteld. Veelal vindt testen plaats met behulp van gerichte invoer van gegevens (testgevallen). Hiervoor zijn diverse testtechnieken beschikbaar. Testen kan echter ook gebeuren door middel van het bijhouden van statistieken tijdens het testproces en/of door een beoordeling van de genomen maatregelen op basis van een checklist.

Het resultaat van de eerste twee stappen worden vaak weergegeven in een zogenaamde strategiematrix (zie figuur 1). Deze tweedimensionale matrix geeft aan welke kwaliteitskarakteristieken 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 kwaliteitskarakteristieken en de drie deelsystemen zijn beschreven in de strategiematrix. Een "x" betekent dat het kwaliteitskarakteristiek van toepassing is op het betreffende deelsysteem, een "xx" betekent dat de combinatie kwaliteitskarakteristiek/deelsysteem speciale aandacht vergt, hetgeen dient te worden vertaald in meer diepgaande tests. Nogmaals wordt benadrukt dat het zinloos is om een bepaalde mate van wiskundige nauwkeurigheid te ontleen aan deze matrix.

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

Figuur 1 : "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 kunde van het betrokken testmanagement. De voornaamste tekortkoming is echter gelegen in het feit dat het communicatiemiddel de (technische georiënteerde) kwaliteitskarakteristieken zijn en de gebruikersbehoeften ten aanzien van kwaliteit geen expliciete rol spelen bij het bepalen van de teststrategie. Het proces van strategie bepaling 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 kwaliteitskarakteristieken. Als tevens geen gebruik gemaakt wordt van standaards, zoals bijv. ISO 9126, inzake de definitie van kwaliteitskarakteristieken, hetgeen vaak het geval is, dan neemt de (spraak)verwarring

met de gebruikers nog verder toe. Desalniettemin wordt geprobeerd tijdens het testen de "fitness-for-use" vast te stellen!

Binnen de produktgerichte benadering van softwarekwaliteit zijn recentelijk een aantal concepten ontwikkeld die kunnen bijdragen tot een grotere gebruikers- en bedrijfsoriëntatie bij het testen van softwareproducten. Achtereenvolgens wordt een beschrijving gegeven van de ontwikkeling ten aanzien van kwaliteitskarakteristieken, van het concept kwaliteitsprofiel en van een methode om de relevante kwaliteitskarakteristieken te identificeren en te specificeren op basis van de kwaliteitsbehoeften van de gebruikers en de bedrijfssituatie.

CONCEPTEN VOOR PRODUKTKWALITEIT

Kwaliteitskarakteristieken

In 1977 kwamen McCall et al. [17] met het voorstel om het concept kwaliteit onder te verdelen in een aantal kwaliteitskarakteristieken. Dit voorstel is overgenomen door een groot aantal andere auteurs, die getracht hebben de kwaliteit van softwareproducten te beschrijven aan de hand van een verzameling van kwaliteitskarakteristieken met daaraan gerelateerde subkarakteristieken. Aan deze subkarakteristieken zijn vervolgens diverse indicatoren en metrieken gekoppeld. Op deze wijze kwamen de diverse auteurs tot een aantal hiërarchische modellen van softwarekwaliteit. In de verschillende modellen komen bepaalde "basis" kwaliteitskarakteristieken steeds weer terug, hoewel hun plaats in de hiërarchie niet steeds dezelfde is. Sommige kwaliteitsexperts zijn van mening dat deze modellen niet gerechtvaardigd kunnen worden omdat er geen universele concepten bestaan waarvan een model voor softwareproduktkwaliteit kan worden afgeleid [16,18]. Deze kritiek op de wetenschappelijke onderbouwing van kwaliteitskarakteristieken ten spijt, is de praktische toepasbaarheid ervan overduidelijk; in de toenemende stroom publikaties over de kwaliteit van softwareproducten lijkt overeenstemming te zijn over de toe te passen kwaliteitskarakteristieken.

De *International Organization for Standardization* (ISO) en de *International Electrotechnical Commission* (IEC) hebben een aantal "standaard" kwaliteitskarakteristieken vastgesteld. Deze standaard kwaliteitskarakteristieken vormen een eerste stap op weg naar consensus in de software industrie en is gericht op een algemene begrip van softwarekwaliteit. De standaard ISO 9126 bevat definities van een zestal kwaliteitskarakteristieken en stelt, in een bijlage bij de standaard, een onderverdeling van de zes kwaliteitskarakteristieken voor in een aantal subkarakteristieken. De standaard is, om spraakverwarring te voorkomen, niet vertaald in het Nederlands, en daarom wordt ook hier de Engelse terminologie gehanteerd. De karakteristieken en voorgestelde subkarakteristieken zijn respectievelijk:

- functionality (functionaliteit), bestaande uit vijf subkarakteristieken: suitability, accuracy, compliance, interoperability en security;
- reliability (betrouwbaarheid), onderverdeeld in de subkarakteristieken 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 subkarakteristieken: analysability, changeability, stability en testability;
- portability (portabiliteit), eveneens bestaande uit vijf subkarakteristieken: adaptability, installability, co-existence, conformance en replaceability.

In werkgroep zes (WG6) van ISO/JTC1/SC7 wordt momenteel nadere studie verricht met betrekking tot indicatoren en metrieken voor elk van de afzonderlijke (sub)karakteristieken [14].

Kwaliteitsprofiel

Het vertrouwen dat een gebruiker, bijv. een afnemer, eindgebruiker of beheerder, in een softwareprodukt stelt, groeit naarmate het testen ervan grondiger is gebeurd. Met grotere testinspanningen zijn uiteraard meer tijd en hogere kosten gemoeid. Echter, zoals eerder bij de beschrijving van de teststrategie aan de orde is gekomen, is het meestal niet nodig om alle kwaliteits(sub)karakteristieken van een softwareprodukt met dezelfde diepgang te testen. Bij bijvoorbeeld een tekstverwerkingsprogramma is het belangrijker bruikbaarheid diepgaand te testen dan betrouwbaarheid. Hetgeen overigens niet betekent dat betrouwbaarheid bij een tekstverwerker totaal

onbelangrijk is. Voor een softwareprodukt dat in een nucleaire installatie wordt gebruikt geldt het omgekeerde: betrouwbaarheid is vele malen belangrijker dan bruikbaarheid. Ook hierbij geldt dat dit niet betekent dat bruikbaarheid totaal onbelangrijk c.q. niet relevant is. Deze nogal intuïtieve aannames zijn nader onderzocht en verder uitgewerkt in het internationale onderzoeksproject SCOPE en werkgroep zes van ISO/JTC1/SC7. Op basis hiervan is een aantal richtlijnen [15] opgesteld, waarin wordt aangegeven hoe diepgaand de evaluatie van een softwareprodukt in een bepaalde situatie dient te zijn. Deze richtlijnen zijn gebaseerd op een onafhankelijke evaluatie van een softwareprodukt in het kader certificatie en zijn derhalve niet specifiek gericht op het testen van softwareprodukten binnen systeemontwikkeling. In deze richtlijnen wordt een viertal *evaluatieneaus* gedefinieerd, met een oplopende mate van diepgang (van niveau D tot en met niveau A). Het vereiste evaluatieniveau voor een bepaalde bedrijfssituatie wordt bepaald aan de hand van factoren zoals economische risico's in geval van discontinuïteit en risico's met betrekking tot veiligheid en beveiliging. De ontwikkelde concepten en richtlijnen zijn met name bedoeld ter ondersteuning bij het vaststellen van een kwaliteitsprofiel. In een kwaliteitsprofiel worden de relevante kwaliteitskarakteristieken van kwaliteit en de bijbehorende evaluatieneaus vastgelegd op basis van bepaalde kenmerken van de bedrijfssituatie. Een kwaliteitsprofiel (zie figuur 2) is in principe een weergave van het vereiste kwaliteitsniveau voor een bepaald softwareprodukt en concretiseert het begrip kwaliteit voor zowel ontwikkelaars, testers als gebruikers.

	Evaluatieniveau				
	-	D	C	B	A
Functionaliteit				X	
Betrouwbaarheid			X		
Bruikbaarheid		X			
Efficiëntie		X			
Onderhoudbaarheid			X		
Portabiliteit	X				

Figuur 2: "Kwaliteitsprofiel"

Een vraag waar tot dusverre echter nog nauwelijks aandacht aan is besteed, is hoe de behoeften van gebruikers ten aanzien van softwarekwaliteit vertaald kunnen worden in een prioriteitenlijst van relevante kwaliteitskarakteristieken c.q. in een kwaliteitsprofiel. Gebruikers zouden moeten proberen hun wensen en behoeften ten aanzien van softwarekwaliteit nader te omschrijven en hierbij, indien noodzakelijk, ondersteuning moeten krijgen. Eveneens geldt dat ontwikkelaars van software moeten proberen de kwaliteit van hun produkten te beschrijven in termen die voor gebruikers duidelijk zijn. Tot dusverre zijn er echter geen systematische benadering, methoden of technieken beschikbaar waarmee zowel gebruikers, ontwikkelaars als testers van software de (vereiste) kwaliteitskarakteristieken van een softwareprodukt eenduidig kunnen bepalen. In de volgende paragraaf wordt ingegaan op de ontwikkeling van een methode voor het bepalen van kwaliteitsprofielen op basis van een analyse van de bedrijfssituatie.

Bepalen van kwaliteitsprofielen op basis van bedrijfsanalyse

In onderzoeksprojecten van de Technische Universiteit Eindhoven (TUE) is onlangs een methode ontwikkeld voor de identificatie en specificatie van kwaliteitskarakteristieken [23]. Deze methode, die wordt toegepast bij KEMA en verder wordt uitgewerkt in de ESPRIT projecten SPACE-UFO en MultiSpace, brengt de wensen en behoeften van gebruikers op basis van hun specifieke bedrijfssituaties in relatie met de kwaliteitskarakteristieken van een softwareprodukt. Gebruikersbehoeften en eisen gesteld vanuit de bedrijfssituatie worden geanalyseerd en nader omschreven met behulp van vragenlijsten. De antwoorden van de gebruikers met betrekking tot hun kwaliteitsbehoeften worden gerelateerd aan de vereiste kwaliteitskarakteristieken door middel van tweedimensionale matrices. De ontwikkelde methode is gebaseerd op het uit de bedrijfskunde afkomstige Process-Control-Information (PCI) model [3]. Volgens het PCI-model kunnen de te stellen eisen met betrekking tot kwaliteit van een softwareprodukt worden bepaald door middel van een analyse van de karakteristieken van een specifieke bedrijfssituatie. Op basis van de geldende literatuur (bijv. [5]) wordt er

onderscheid gemaakt in drie soorten karakteristieken: karakteristieken van bedrijfsprocessen, gebruikerskarakteristieken en karakteristieken van het softwareprodukt.

Karakteristieken van bedrijfsprocessen

Het eerder genoemde PCI-model is gebruikt om de karakteristieken te onderzoeken die van invloed zijn op de beheersing van het bedrijfsproces of bedrijfssysteem. De systeemleer heeft diverse bruikbare concepten aangereikt die vertaald zijn in bedrijfsproceskarakteristieken. Deze karakteristieken zijn vervolgens in relatie gebracht zijn met relevante kwaliteitskarakteristieken. De karakteristieken van een bedrijfsproces zijn:

- het *aantal* procesvariabelen en hun *onderlinge afhankelijkheid*
- de *beheersbaarheid* en *voorspelbaarheid* van procesvariabelen
- de *gevoeligheid* en *stabiliteit* van het proces
- het *reactiepatroon* van het proces.

Er worden twee voorbeelden gegeven van verbanden tussen deze bedrijfsproceskarakteristieken en kwaliteitskarakteristieken (volgens ISO 9126):

- als het aantal procesvariabelen en hun onderlinge afhankelijkheid groot is, dan is het proces moeilijk te beheersen; de behoefte aan usability, d.w.z. understandability en operability is dan groot.
- indien het proces zeer gevoelig is bij een geringe stabiliteit, dan bestaat de behoefte om snel en accuraat te kunnen reageren op een storende omgevingsfactor. In termen van kwaliteitskarakteristieken betekent dit in dat maintainability, bijv. changeability en wellicht ook portability belangrijk zijn.

Gebruikerskarakteristieken

Gebruiker zijn direct dan wel indirect betrokken bij de aankoop, het gebruik of beheer van het softwareprodukt. Derhalve dienen menselijke factoren een grote rol te spelen bij het bepalen en specificeren van de kwaliteitskarakteristieken. Gebruikers kunnen worden onderverdeeld in de volgende categorieën:

- input georiënteerde eindgebruikers
- gebruikers die de informatie uit het systeem gebruiken, zogenaamde output users
- management (gebruikers op afstand), die hun informatie uit het systeem verkrijgen via tussenpersonen.

De gebruikerscategorieën kunnen nader worden gespecificeerd vanuit drie verschillende invalshoeken: de *ervaring* die gebruikers hebben met vergelijkbare softwareproducten en met softwareproducten in het algemeen, het *opleidingsniveau* van de gebruikers en het *aantal* gebruikers.

Ook hierbij wederom een tweetal voorbeelden:

- een softwareprodukt dat bedoeld is voor een groot aantal onervaren gebruikers met een laag opleidingsniveau stelt hoge eisen aan understandability en learnability (beide zijn subkarakteristieken van usability).
- indien sprake is van een groot aantal input georiënteerde eindgebruikers betekent dat het softwareprodukt een hoog gebruiksgemak moet hebben, hetgeen hoge eisen stelt aan operability en time-behaviour.

Softwareprodukt karakteristieken

Softwareproducten kunnen worden gekarakteriseerd door een drietal dimensies: het type softwareprodukt, de wijze van gebruik en de vereiste technische infrastructuur [23]:

- Het *type* softwareprodukt: kan het produkt worden gekarakteriseerd als, bijv. een gegevensverwerkend systeem, een beslissingsondersteunend systeem, een kennissysteem, of een tekstverwerker? Kan het softwareprodukt worden gekarakteriseerd als een standaard pakket of een maatwerk systeem voor een specifieke afnemer?
- De *wijze van gebruik*: hoeveel uur per dag wordt is het softwareprodukt operationeel, dient het softwareprodukt te worden gekarakteriseerd als een on-line of een batch georiënteerd systeem en worden batch-jobs tevens gedurende de dag gedraaid?
- De vereiste *technische infrastructuur*: is het softwareprodukt vervaardigd om te kunnen functioneren op verschillende soorten hardware platforms en biedt het de mogelijkheid om operationeel te zijn in een netwerkomgeving?

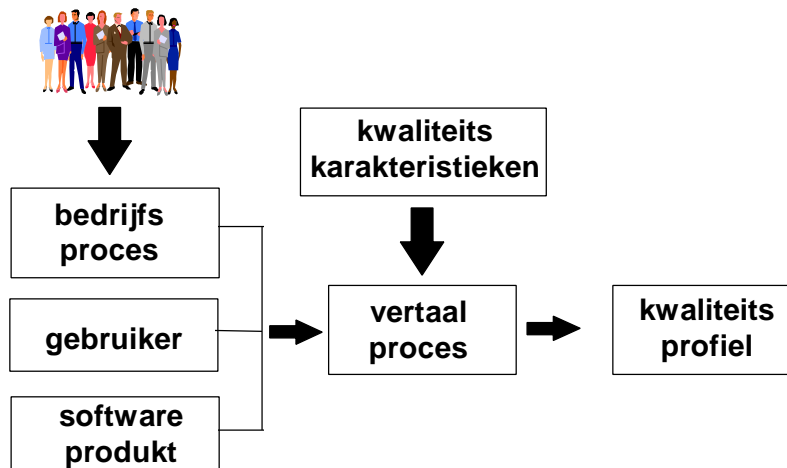
Twee voorbeelden met betrekking tot softwareprodukt karakteristieken:

- bij een gegevensverwerking systeem met een groot aantal batch-jobs is de recoverability (een subkarakteristiek van reliability) waarschijnlijk van groot belang.
- als het softwareprodukt wordt gekarakteriseerd als een standaard pakket is portabiliteit veelal een belangrijke kwaliteitskarakteristiek.

Het merendeel van bovengenoemde relaties is vooralsnog in hoofdzaak intuïtief en hypothetisch van aard. Hoewel in een pilotprojecten reeds een aantal zeer goede resultaten zijn geboekt, moeten de relaties nog verder uitgewerkt en in de praktijk worden gevalideerd.

Evaluatiemodel

In deze paragraaf wordt een evaluatiemodel beschreven op basis waarvan de kwaliteitskarakteristieken van een softwareprodukt kunnen worden vastgesteld. Door middel van dit model proberen we het proces van het bepalen van de relevante kwaliteitskarakteristieken van een softwareprodukt meer persoonsonafhankelijk c.q. minder expert afhankelijk maken. Het voornaamste doel hiervan is een herhaalbaar en reproduceerbaar proces te creëren waarin de kwaliteitsbehoeften van gebruikers en de karakteristieken van de bedrijfssituatie worden vertaald in de relevante kwaliteitskarakteristieken en corresponderende evaluatieniveaus. Het model is weergegeven in figuur 3.



Figuur 3: "Evaluatiemodel"

Het vertaalproces bestaat uit een aantal stappen waarin de kwaliteitsbehoeften van de gebruikers en de karakteristieken van de bedrijfssituatie worden vertaald in een aantal goed gespecificeerde kwaliteitskarakteristieken. Communicatie met vertegenwoordigers uit de gebruikersorganisatie staat hierbij centraal. Tijdens de eerste stap wordt gebruik gemaakt van gestructureerde vragenlijsten ter bepaling van de kwaliteitsbehoeften van de gebruikers ten aanzien van een softwareprodukt. Tijdens de tweede stap worden de antwoorden die de gebruikers hebben gegeven op (semi-)deterministische wijze gerelateerd aan kwaliteitskarakteristieken door middel van tweedimensionale matrices. Tijdens de derde en tevens laatste stap wordt in samenwerking met de gebruikers de betekenis van de kwaliteitskarakteristieken in de context van de bedrijfssituatie bepaald, evenals het relatieve belangrijk van de diverse kwaliteitskarakteristieken. Tijdens deze stap wordt het kwaliteitsprofiel uitgebreid met een grote hoeveelheid waardevolle context-informatie. Informatie die aangeeft waarom bepaalde kwaliteitskarakteristieken wel en anderen niet van belang zijn.

HET KWALITEITSPROFIEL ALS UITGANGSPUNT VOOR TESTERS

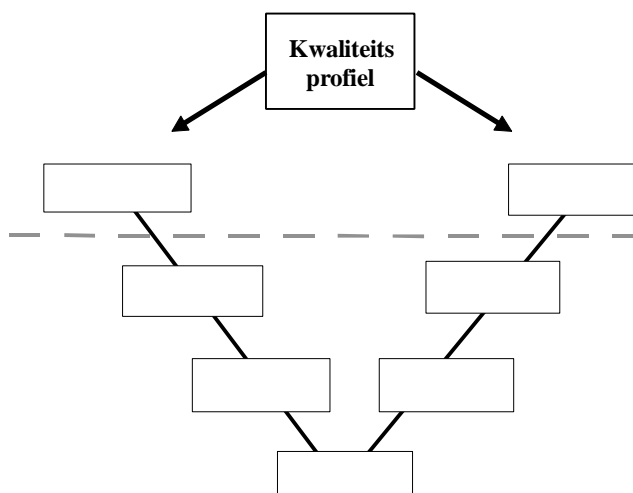
Binnen het evaluatiemodel komt een kwaliteitsprofiel tot stand dat gebaseerd is op de kwaliteitsbehoeften van de gebruikers en derhalve kan worden gekenschetst als een gebruikers georiënteerd kwaliteitsprofiel. Een vergelijking van het kwaliteitsprofiel met de strategiematrix, levert een groot aantal overeenkomsten. Beiden geven aan welke kwaliteitskarakteristieken belangrijk zijn in een specifieke (bedrijfs)situatie en in beide komt

het belang van de geselecteerde kwaliteitskarakteristieken tot uiting. Bij het kwaliteitsprofiel wordt dit bereikt door middel van de evaluatieniveaus en bij de strategiematrix door middel van het relatieve belang. In het proces van het bepalen van de relevante kwaliteitskarakteristieken ligt in beide methoden de nadruk op een intensieve communicatie met gebruikers. Het grootste verschil tussen de twee methoden is gelegen in het feit dat het kwaliteitsprofiel, wanneer het toegepast wordt bepaald met behulp van het evaluatiemodel, daadwerkelijk gebruikers georiënteerd is. Een ander verschil is het ontbreken van een expliciete risico-analyse binnen het evaluatiemodel, hoewel er wel een diepgaand context onderzoek plaatsvindt op basis van de karakteristieken van de bedrijfssituatie. Dit betekent dat indien het kwaliteitsprofiel wordt gebruikt als basis voor het testen van een softwareprodukt, een aantal (kleine) aanpassingen noodzakelijk zijn op basis van een uitgevoerde risico-analyse 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 gebruikers georiënteerde teststrategie. Dit resulteert vervolgens in een meer gebruikers georiënteerd testproces. Door middel van de uitgevoerde test wordt het mogelijk een uitspraak te doen over de “fitness for use” van het softwareprodukt!

Kwaliteitsprofiel ook voor ontwikkeling

Opgemerkt dient te worden dat het gebruik van het evaluatiemodel en het kwaliteitsprofiel niet enkel van toepassing kan zijn bij het testen van softwareproducten. Het kan worden gebruikt binnen het systeemontwikkelingsproces. In combinatie met de informatie over de karakteristieken van de bedrijfssituatie kan het kwaliteitsprofiel toegepast worden om te komen tot gefundeerde specificaties van kwaliteitseisen. Het proces ter bepaling van een kwaliteitsprofiel op basis van het evaluatiemodel wordt in het ideale situatie uitgevoerd door ontwikkelaars, tester en gebruikers tezamen.



Figuur 4: “Kwaliteitsprofiel en het V-model”

CONCLUSIES EN VERDER ONDERZOEK

Pilotprojecten hebben aangetoond dat het evaluatiemodel een goed hulpmiddel is bij de bepaling van een kwaliteitsprofiel. Op basis van de kenmerken van het bedrijfsproces, de gebruikers en het softwareprodukt kunnen de gewenste kwaliteitskarakteristieken worden vastgesteld. Vervolgens kan een kwaliteitsprofiel worden gebruikt als basis voor een teststrategie, die als uitgangspunt dient bij het beoordelen van de kwaliteit van het softwareprodukt. Op deze wijze worden de gebruikersbehoeften met betrekking tot kwaliteit een expliciet onderdeel van het testproces, hetgeen leidt tot een meer gebruikers georiënteerd testproces.

Verder onderzoek bij de TUE, KEMA en in de genoemde ESPRIT projecten zal zich richten op een nadere uitwerking van het evaluatiemodel, de evaluatieniveaus en een verdere concretisering van het testproces. De diverse onderzoeksrichtingen worden hieronder kort toegelicht:

Uitwerking van het evaluatiemodel

Voor ieder van de drie soorten bedrijfskarakteristieken: bedrijfsproces, gebruikers en softwareprodukt, dienen de vragenlijsten nader te worden uitgewerkt en in de praktijk gevalideerd. Op basis van de antwoorden die de gebruikers geven op de gestelde vragen moet het, in de toekomst, mogelijk zijn op deterministische wijze een lijst van geprioriteerde kwaliteitskarakteristieken op te stellen voor een specifieke bedrijfssituatie.

De evaluatieniveaus

Het werken met evaluatieniveaus vereist dat er normen en standaards en daaraan gerelateerde metrieken voorhanden zijn. Zonder dergelijke normen blijft de bepaling van een evaluatieniveau beperkt tot een subjectief oordeel. Verder onderzoek zal zich richten op de ontwikkeling van objectieve normen voor de diverse evaluatieniveaus.

Concretisering van het testproces

Er bestaat nog steeds relatief weinig kennis over de mogelijkheden tot het beoordelen van softwarekwaliteit. Met name ten aanzien van de relaties tussen soorten softwareproducten, kwaliteitskarakteristieken, indicatoren (acceptatiecriteria) en metrieken moet nog veel onderzoek worden verricht. Het onderzoek heeft tot doel de ontwikkeling van een typologie van softwareproducten, met voor iedere type softwareprodukt een lijst van relevante kwaliteitskarakteristieken vergezeld van indicatoren, metrieken en testtechnieken.

Bovengenoemde onderzoekswerkzaamheden zullen op relatief korte termijn plaatsvinden met als doel een gedegen en praktische methode voor het testen van de softwarekwaliteit op basis van de kwaliteitsbehoeften van gebruikers en de eisen die worden gesteld vanuit de bedrijfssituatie. Voor KEMA is het uiteindelijke doel een objectief, effectief en efficiënt evaluatieproces met betrekking tot softwareproducten uit te kunnen uitvoeren in het kader van een certificeringsonderzoek.

LITERATUUR

1. Bache R. and G. Bazzana, *Software metrics for product assessment*, McGraw-Hill, London, 1994.
2. Basili V.R., J.D. Musa, *The future engineering of software: a management perspective*, IEEE Computer, Vol. 24, No. 9, 1991.
3. Bemelmans T.M.A., *Bedrijfskundig Ontwerpen van Bestuurlijke Informatiesystemen*, in: P.A. Cornelis, J.M. van Oorschot (eds.), *Automatisering met een menselijk gezicht*, Kluwer, 1986.
4. Boehm B.W. et al, *Characteristics of Software Quality*, TRW Series of Software Technology, Vol. 1, North Holland Publishing Company, 1978.
5. Davies G.B., M.H. Olsen, *Management Information Systems*, McGraw-Hill, London, 1984.
6. Delen G.P.A.J., D.B.B. Rijsenbrij, *The Specification, Engineering and Measurement of Information Systems Quality*, in: *Journal System Software*. Vol. 17, 1992.
7. Dorling A., *SPICE: Software Process Improvement and Capability dEtermination*, in: *Software Quality Journal*, Vol. 2, 1993.
8. Heemstra F.J., R.J. Kusters, J.J.M. Trienekens, *Van kwaliteitsbehoeften naar kwaliteitseisen*, Academic services, 1996.
9. Humphrey W.S., *Managing the Software Process*, Addison-Wesley, 1989.
10. Information Technology Security Evaluation Criteria (ITSEC), *Provisional harmonised criteria, version 1.2*, Commission of the EC, 1991.
11. ISO 8402, *Quality management and quality assurance - Vocabulary*, International Organization of Standardization, 1994.
12. ISO 9000-3, *Quality management and quality assurance standards, Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software*, International Organization of Standardization, 1991.
13. ISO/IEC 9126, *Information Technology - Software product evaluation - Quality characteristics and guidelines for their use*, International Organization of Standardization, 1991.

14. ISO/IEC/JTC1/SC7 N1201, *Working Draft; Information Technology: Indicators and Measures*, ISO/IEC, 1994.
15. ISO/IEC/JTC1/SC7 CD 14598-5, *Information Technology - Software product evaluation - Part 5: Evaluators guide*, International Organization of Standardization, 1994
16. Kaposi A. and M. Myers, *Systems, models and measures*, Springer-Verlag, London, 1994
17. McCall, J.A., P.K. Richards and G.F. Walters, *Factors in Software Quality*, RADC-TR-77-363 Rome Air Development Center, Griffis Air Force, Rome, NY, 1977.
18. NGGO, *Over het testen van informatiesystemen*, Tutein Nolthenius, 1991.
19. Pol, M., *Reliable systems require structured test strategy*, in: Newsletter of the Institute of Internal Auditors Benelux, Vol. 9, September 1992.
20. Pol, M, R.A.P. Teunissen and E.P.W.M. van Veenendaal, *Testen volgens TMap*, Tutein Nolthenius, 1995.
21. SERC-Quint, *Het specificeren van software-kwaliteit: een praktische handleiding*, Kluwer Bedrijfswetenschappen, 1992.
22. TickIT, *Guide to Software Quality Management Systems, Construction and Certification using EN 29001*, TickIT Office, London, 1991
23. Trienekens J.J.M., *Tijd voor kwaliteit: werken aan betere informatiesystemen*, Thesis Publishers, Amsterdam, 1994.
24. Trienekens J.J.M., *Quality Management in Software Production, A Customer Oriented Approach*, in: Proceedings of the IFIP 5.7 Working Conference on Integration in Production Management, H.J. Pels and J.C. Wortmann (eds.), The Netherlands, North-Holland, 1992.
25. Veenendaal, E.P.W.M. van, *Het invoeren van gestructureerd testen*, in: Informatie management, Augustus 1993.
26. Zeist, B. van, P. Hendriks, R. Paulussen en J. Trienekens, *Kwaliteit van softwareproducten: praktijkervaringen met een kwaliteitsmodel*, Kluwer Bedrijfswetenschappen, 1996

Over de auteurs

Drs. Erik van Veenendaal is de oprichter en directeur van Improve Quality Services. Improve Quality Services biedt services op het gebied van kwaliteits management, usability en testen. Tevens is hij verbonden aan de TU-Eindhoven in het kader van promotie-onderzoek met betrekking tot het testen van softwareproducten.

Dr. Jos Trienekens is universitair hoofddocent en leidt praktijkgerichte IT-onderzoeksprojecten bij het Frits Philips Institute for Quality Management (FPIQM) en de TU-Eindhoven, Faculteit Technologie Management. Hij is voorzitter van ENCRESS Nederland (European Network for Clubs on Reliability of Software-intensive systems). Hij is als consultant betrokken bij de uitvoering van het ESPRIT project SPACE-UFO.