Doelgericht beoordelen van software

Doelgericht beoordelen van software

Teade Punter

Technische Universiteit Eindhoven

CIP-DATA BIBLIOTHEEK TECHNISCHE UNIVERSITEIT EINDHOVEN

Punter, Hendrik Teade

Doelgericht beoordelen van software / door Hendrik Teade Punter. - Eindhoven : Technische Universiteit Eindhoven, 2001. – Proefschrift. -
ISBN 90-386-0863-2
NUGI 684; 855

Trefwoorden: Software; Software-kwaliteit; Software-evaluatie; Softwarebeheer; ISO 9126; ISO 14598; GoalQuestionMetric; GQM

Druk: Universiteitsdrukkerij Technische Universiteit Eindhoven

Doelgericht beoordelen van software

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de Rector Magnificus, prof.dr. M. Rem, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen op
donderdag 8 maart 2001 om 16.00 uur

door

Hendrik Teade Punter

geboren te Oosterwolde

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. T.M.A. Bemelmans
en
prof.dr.ir. A.C. Brombacher

Copromotor:

dr.ir. J.J.M. Trienekens

Voor Dieuwke

Voorwoord

Dit proefschrift is het resultaat van promotie-onderzoek dat ik vanaf april 1996 heb uitgevoerd bij de capaciteitsgroep Informatie en Technologie van de faculteit Technologie Management van de Technische Universiteit Eindhoven. Het onderzoek was onderdeel van het Corporate Research Development programma 'Software Product Quality' van KEMA Nederland B.V te Arnhem. Het onderzoek is verricht in de onderzoeksschool Beta.

Voor dit onderzoek ben ik geïnspireerd, begeleid en bijgestaan door diverse mensen. Allereerst bedank ik de leden van de kerncommissie voor de begeleiding van het onderzoek. Het vertrouwen dat Jos Trienekens gedurende het onderzoek in mij stelde waardeer ik zeer. Hierdoor kreeg ik de ruimte om een eigen richting aan het onderzoek te geven. Rob Kusters ben ik erkentelijk voor zijn rol als aandrijver van enerverende discussies. Theo Bemelmans waardeer ik voor zijn direct aanpak en de les dat de vorm en inhoud van een proefschrift niet te scheiden zijn. Aarnout Brombacher bedank ik voor zijn kritische maar immer positieve inbreng tijdens het project.

Vier oud-collega's bij de Open Universiteit (OU) hebben mij gestimuleerd tot het doen van onderzoek. Fred Mulder en Karel Lemmen bedank ik voor de ruimte die ze mij gaven om onderzoek te doen. Fred Heemstra en Paul Hendriks (via de samenwerking tussen OU en Serc) ben ik erkentelijk voor de introductie in de wereld van software management en metrieken.

Tijdens het onderzoek hebben diverse mensen mij geïnspireerd met bestaande en nieuwe ideeën over beoordelen, meten, software-kwaliteit en onderzoek doen. Ik bedank hiervoor met name: Rini van Solingen, Peter Eisinga, Geert Poels, Tom Dolan, Paul Bagchus, Michiel van Genuchten, Bruno Peeters, Frank Niessink, Frank Simon en Michael O'Duffy.

Het type onderzoek dat ik verricht heb vereist toepassing in en voeding vanuit de praktijk. Allereerst bedank ik de mensen bij KEMA Nederland voor de mogelijkheid om mee te werken aan hun onderzoeksprojecten en opdrachten. Peter van Sprundel, Gerard Duin, Jaap van Ekris en Marja Visser waren hierbij prettige collega's. Het onderzoek is ook uitgevoerd bij andere bedrijven. Ik bedank Ton van Opstal (ABP), Alexander Westra (Cap Gemini ISM), Erik Rodenbach (Tokheim RPS), Mans Mesken (Emendo) en Erik van Veenendaal (Océ Technologies, Improve QS) voor de mogelijkheden die ze mij hierbij hebben geboden. De samenwerking in de werkgroep Software metrieken van Nesma/Nggo, het EuroScope-consortium en het Esprit Space-Ufo-project heeft het onderzoek een bredere basis gegeven.

Een speciaal woord van waardering gaat uit naar mijn kamer- en ganggenoten op de universiteit: Rini van Solingen, Mark van der Zwan, Erik van Veenendaal, Sjaak Bouman, Frank Berkers (en later) Laura Maruster, Christine Pelletier en Nick Szirbik: it was a great

pleasure to be your roommate. Furthermore thanks to all my colleagues of the capacity group Information and Technology in Eindhoven as well as Fraunhofer IESE in Kaiserslautern.

Verder ben ik Theo en Janny Punter-de Vries en Roelof en Mechtelien de Haan-Jöbsis zeer erkentelijk voor hun stimulerende belangstelling tijdens het onderzoek. De eindredactie, uitgevoerd door Frouke Punter en Roelof de Haan, heeft het proefschrift verbeterd.

Tenslotte bedank ik Dieuwke, mijn echtgenote. Allereerst voor de vakinhoudelijke discussies die we over het onderzoek hebben gehad. De zondagochtenden, wandelend langs de Geul, waren een geëigend moment voor kritische vragen en leverden veelal ideeën op over de te volgen strategie. Met de geboortes van Karsten en Wieger veranderden de momenten waarop aan het onderzoek kon worden gewerkt. Voor de manier waarop we een balans vonden tussen ons beider werk en de opvoeding van onze beide zoons ben ik Dieuwke zeer erkentelijk. Daarom is het proefschrift aan haar opgedragen.

Teade Punter

Kaiserslautern (D), 1 december 2000

Inhoudsopgave

1. Inleiding en probleemstelling

Dit proefschrift gaat over het beoordelen van softwareproducten. In dit eerste hoofdstuk wordt de probleemstelling rondom dergelijke beoordelingen geformuleerd. Daartoe worden eerst enkele relevante begrippen geïntroduceerd.

1.1 Softwareproduct-kwaliteit

Deze paragraaf gaat in op belangrijke begrippen voor het beoordelen van softwareproducten. Eerst wordt gedefinieerd wat er in dit onderzoek onder softwareproduct wordt verstaan (1.1.1). Vervolgens komt aan de orde dat het bij beoordelingen gaat om het beoordelen van de kwaliteit van de producten (1.1.2). Tenslotte komt het belang van de beoordelingen aan de orde (1.1.3).

1.1.1 Softwareproduct

Wat is een softwareproduct?
Een *softwareproduct* wordt in dit proefschrift gezien als een bepaald soort ICT-toepassing. Het kan hierbij gaan om verschillende toepassingen. Een toepassing kan op zich zelf staan, het kan ook onderdeel van een ander product of informatiesysteem zijn. Voorbeelden van een softwareproduct waarbij de software een relatief zelfstandig product vormt zijn: de NS-reisplanner, tekstverwerkers en computerspelletjes. Voorbeelden waarbij software wordt toegepast in een ander product zijn: mobiele telefoons, kopieermachines, scanapparatuur voor de zorgsector en het Automatische Trein Beveiligingssysteem (ATB) voor het regelen van de snelheid van treinen. Voorbeelden waarbij software wordt toegepast in een informatiesysteem zijn: een orderverwerkingssysteem van een postorderbedrijf of een back-office systeem van een verzekeringsmaatschappij voor het afsluiten van levensverzekeringen.

Er is een aantal begrippen dat sterk verwant is met softwareproduct. Het eerste begrip is *software component:* ‘This is an identifiable and self-contained portion of a software product’ (ISO 14598-5, 1998). Voorbeelden van componenten zijn: protocollen voor communicatiebeveiliging, delen van toepassingsapplicaties. Componenten worden toegepast in andere componenten of zijn onderdeel van een softwareproduct of informatiesysteem (Syperski, 1998).

Een tweede aan softwareproduct gerelateerde begrip is *software package:* ‘This is a complete and documented set of programs supplied to several users for a generic application or function’ (ISO 2382-1, 1993). Voorbeelden van software packages zijn: de MS-Word-

tekstverwerker en Enterprise Resource Planning (ERP) pakketten, zoals geproduceerd door SAP en Oracle.

Dit proefschrift richt zich qua voorbeelden in het bijzonder op softwareproducten waarbij de software in een ander product wordt gebruikt –zogenaamde embedded softwareproducten– en producten waarbij de software onderdeel is van een informatiesysteem. Het kan hierbij gaan om een specifieke component, maar ook om een softwarepakket. Een embedded softwareproduct definiëren we exacter als: 'software that determines the functionality of microprocessors and other programmable devices that are used to control electronic, electrical and electromechanical equipment and sub-systems. The programmable devices are often 'invisible to the user' (Tickit, 1995). Er is een heel scala aan embedded softwareproducten op de markt. In dit proefschrift gaat het wat betreft de voorbeelden om embedded software voor kopieerapparaten, kassa registersystemen en medische apparatuur.

De tweede categorie betreft software die wordt toegepast in informatiesystemen. Bij een informatiesysteem ligt de nadruk op de informatievoorziening. Er wordt informatie verzameld, opgeslagen en gedistribueerd. Informatiesystemen zijn op verschillende manieren te karakteriseren, bijvoorbeeld op basis van de dimensie toepassingsgebied[1]. In dit proefschrift gaat het qua voorbeelden om informatiesystemen voor huisartsen en voor een verzekeringsmaatschappij.

Onderwerpen van een softwareproduct

Na de introductie van hetgeen er in dit proefschrift onder softwareproducten wordt verstaan, komt de vraag aan de orde 'wat er tot een softwareproduct wordt gerekend?' Hiervoor geven we eerst de ISO 12207 (1995) definitie van softwareproduct. Deze ISO-standaard definieert softwareproduct als 'the set of computer programs, procedures, and possibly associated documentation and data.

Programs worden in deze definitie als eerste onderwerp van een softwareproduct onderkend. Program betreft de (source)code die de functionaliteit van het product bepaalt. Het kan ook gaan om een interface met andere producten of systemen, zoals de Application Programming Interface (API) in MicroSoft software.

Het als tweede onderkende onderwerp zijn de *procedures*. Deze beschrijven de wijze om het product te gebruiken en te onderhouden. Onder procedures kan in feite ook de documentatie worden begrepen. De *documentatie* –volgens voorgaande ISO-definitie als derde onderdeel gedefinieerd- beschrijft namelijk hoe de gebruiker het product dient toe te passen (user documentation). Ook kan het de structuur of de ontwerpoverwegingen voor de ontwikkelaars beschrijven (technical documentation). Met het beoordelen van procedures

[1] ISO CD 12182 (1994) onderkent zelfs vijftien verschillende dimensies om informatiesystemen te rubriceren, waaronder: application area of IS, scale of software, software criticality, user class en required performance.

komt een productbeoordeling in de buurt van een proces assessment (zie ook paragraaf 1.2.2.).

Gegevens (data) zijn het vierde onderwerp dat de ISO-definitie aan een softwareproduct onderkent. Het gaat dan om de beschrijving van gegevens zoals gebruikt door het product, bijvoorbeeld de data ingegeven door de gebruikers of data om het programma te laten functioneren.

Naast de voorgaande vier onderwerpen van de ISO-definitie, worden in de beoordelingspraktijk soms ook de service en de hardware tot het softwareproduct gerekend. Bij service gaat het dan om de dienstverlening voor het oplossen van problemen die optreden bij het product, bijvoorbeeld de problemen die tijdens het installeren van de software ontstaan. Hardware betreft het platform waarop de software moet worden geïnstalleerd. Hardware wordt soms tot een softwareproduct gerekend, vanuit het idee dat de software niet functioneert zonder een (personal) computer.

1.1.2 Softwareproduct kwaliteit

Bij het beoordelen van softwareproducten richten we ons op het beoordelen van de kwaliteit van die producten. De kwaliteit van iets en dus ook van softwareproducten is niet eenduidig te definiëren. Wat er onder kwaliteit wordt verstaan hangt af van het perspectief waarmee naar de software wordt gekeken. Met andere woorden: ‘quality is in the eyes of its beholders’ (Fenton en Pfleeger, 1996) of kwaliteit van software is ‘highly context dependent ... and no universal definitions of quality exist’ (Kitchenham en Pfleeger, 1996).

In de literatuur over software-kwaliteit worden vaak vijf perspectieven op kwaliteit onderkend. Deze door Garvin (1984) onderkende perspectieven definiëren elk op een eigen manier software(product)kwaliteit. De onderkende perspectieven zijn:

- Transcendent based – kwaliteit is universeel herkenbaar. Men ervaart het eenvoudigweg of in het geheel niet.
- Product based – kwaliteit komt tot uiting in een verzameling van inherente karakteristieken van een product.
- User based – kwaliteit is wat de gebruiker ervan vindt. Dit perspectief wordt vaak verwoord als ‘quality is fitness for purpose’ of ‘quality is fitness for use’.
- Manufacturing based – kwaliteit heeft betrekking op het voldoen van een product aan vooraf opgestelde specificaties. Dit perspectief wordt vaak verwoord als ‘conformance to specifications’.
- Value based – kwaliteit is niet absoluut maar wordt bekeken in relatie tot tijd, kosten en inspanning, nodig om een bepaald kwaliteitsniveau te behalen. Dit perspectief wordt vaak verwoord als: ‘quality is value for money’.

Het ‘transcendent based’ perspectief leidt niet tot een definitie van wat er onder de kwaliteit van een specifiek product wordt verstaan. Het perspectief gaat ervan uit dat het duidelijk is

of een product goed is of niet. Het perspectief is moeilijk bruikbaar voor het beoordelen van softwareproductkwaliteit, omdat men ervan uitgaat dat deze kwaliteit niet expliciet kan worden gemaakt.

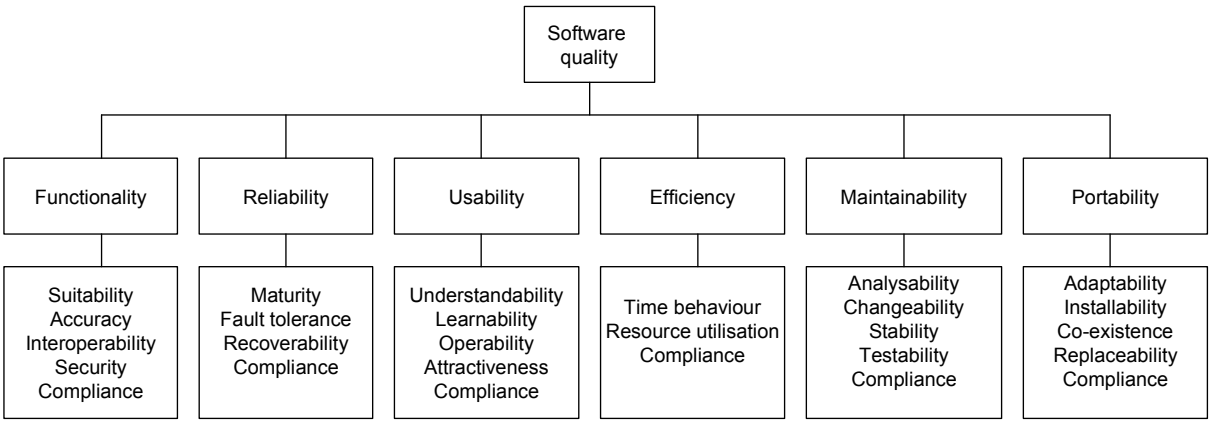
Het 'product based' perspectief gaat ervan uit dat verschillen in de kwaliteit van producten zijn terug te voeren op verschillen in kenmerken van een product. Grof gezegd: 'een product met meer kenmerken heeft meer kwaliteit'. Dit perspectief is in het verleden vaak toegepast op softwareproducten, zie bijvoorbeeld Boehm e.a. (1978) en (Deutsch en Willis, 1988). De kritiek op dit perspectief is dat meer kenmerken aan het product ook tot hogere kosten zal leiden (Trienekens, 1994). Volgens het 'value based' perspectief zijn kosten echter een belangrijke kwaliteitkenmerk. Kwaliteit vanuit het 'product based' perspectief conflicteert hiermee met kwaliteit vanuit het 'value based' perspectief.

Het 'user based' perspectief heeft de laatste jaren aan belang gewonnen. Dit komt door het inzicht in zowel de literatuur als de praktijk dat het uiteindelijk de gebruikers van het softwareproduct zijn, die de kwaliteit van een product bepalen. Het belang van het 'user based' perspectief blijkt bijvoorbeeld uit het feit dat er een apart Esprit onderzoeksproject is gestart om een methode op te stellen waarmee softwareproductkwaliteit vanuit het perspectief van de gebruiker kan worden gedefinieerd (Space Ufo consortium, 1998).

Het 'manufacturing based' perspectief speelt van oudsher een belangrijke rol bij het bepalen van softwareproductkwaliteit. Men houdt zich vast aan vooraf gespecificeerde eisen en meet tijdens het ontwikkelproces of het product voldoet aan die gestelde eisen. Minder aan de orde is of die 'produktie-eisen' ook werkelijk een afspiegeling zijn van wat eindgebruikers uiteindelijk willen en dat de wensen van gebruikers veranderen tijdens het productieproces.

Hoewel softwareproduct kwaliteit vanuit ieder van de perspectieven afzonderlijk is te beschouwen menen wij dat voor een adequate beschrijving veelal een combinatie van perspectieven nodig is. In de literatuur en praktijk gebeurt dat ook. Een voorbeeld is de definitie die ISO 8402 (1994), ISO 9126 (1991) en ISO 14598-1 (1999) als standaarden voor (software)kwaliteit geven: 'the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs'. Hier is sprake van een combinatie van zowel het 'product based' als het 'user based' perspectief. Het eerste perspectief blijkt uit de zinsnede 'totality of characteristics', het tweede uit de zinsnede: 'to satisfy stated and implied needs'.

Voor het beschrijven van software (product)kwaliteit zijn vele kwaliteitsmodellen opgesteld. Voorbeelden zijn (Boehm e.a., 1978), (Rocha en Palermo, 1989), (Kaposi en Kitchenham, 1987), (Serc, 1992), (ISO 9126, 1991), (Delen en Rijsenbrij, 1990a, 1990b), (Dromey, 1996). Tegenwoordig wordt veel naar het ISO 9126-kwaliteitsmodel verwezen. Dit model definieert software-kwaliteit aan de hand van zes hoofd-karakteristieken. Elk wordt onderverdeeld in subkarakteristieken, 27 in totaal. Onderstaande figuur geeft een overzicht van de structuur.



Figuur 1.1 Het ISO 9126-kwaliteitsmodel (ISO CD 9126, 1999)

Kritiek op het ISO 9126-kwaliteitsmodel wordt gegeven door Kitchenham e.a. (1995b), Mellor (1992) en Dromey (1996). De kritiek betreft onder andere de definitie van Reliability en de decompositie van de karakteristieken.

Het ISO 9126-kwaliteitsmodel –en ook de andere modellen- moeten worden opgevat als een raamwerk. De modellen zijn hulpmiddelen om kwaliteit te definiëren. Maar er kan van worden afgeweken. Dit gebeurt als men bijvoorbeeld met de door ISO 9126 opgestelde definitie van Reliability niet uit de voeten kan. Wat dit betreft is Fenton en Pfleeger's (1996) uitspraak van toepassing als zij stellen 'that it will be impossible to define the ultimate quality model'.

1.1.3 Waarom de kwaliteit van softwareproduct beoordelen?

Het maatschappelijk belang van softwareproducten is de afgelopen decennia enorm gegroeid. Het functioneren van bijvoorbeeld het telefoonnetwerk, de aandelenbeurs en medische apparatuur is tegenwoordig in belangrijke mate afhankelijk van software. Als deze software niet functioneert volgens verwachtingen, dan kost dat tijd, geld, ergernis en soms zelfs mensenlevens.

Er zijn vele gevallen van slechte softwareproducten bekend. Illustratief is het falen van software in vliegtuigen en raketten. Een voorbeeld is de mislukte lancering van de Ariane V rocket in 1996 (Lions, 1996). Dergelijke ongelukken benadrukken het belang van goede software omdat ze dramatische gevolgen hebben: het falen van de software leidt tot doden en gewonden en daarnaast tot een enorme financiële strop. De gevolgen van slechte software hoeven overigens niet altijd dramatisch te zijn. Zo kan het zijn dat een bepaalde functie niet geactiveerd kan worden. Een voorbeeld is het niet kunnen activeren van een door de producent aangeprezen mogelijkheid om e-mail te verzenden via een mobiele telefoon. De hoofdfunctionaliteit -het voeren van gesprekken-, blijft in dit voorbeeld weliswaar overeind, maar een deel van de functionaliteit vervalt door slecht werkende software.

Voorgaande voorbeelden illustreren het belang van softwareproductkwaliteit. Binnen de software engineering zijn in de loop der tijd verschillende initiatieven ontplooid om die kwaliteit te beheersen en te verbeteren. Voorbeelden zijn: het toepassen van nieuwe programmeertalen (Meijer, 1988), het introduceren van inspecties en reviews in het ontwikkelingsproces (Gilb en Graham, 1990), toepassen van gestructureerd ontwikkelen (Yourdon, 1989) en het gebruik van Case-tools (Reeken en Trienekens, 1991).

Dergelijke initiatieven zijn belangrijk om kwalitatief goede software te bouwen. We constateren echter ook dat ze geen garantie geven dat software-kwaliteit daadwerkelijk wordt gerealiseerd. Daarvoor zal men softwareproducten moeten beoordelen. We beschouwen het beoordelen dan ook als onderdeel van het streven om kwaliteit te beheersen en te verbeteren. Het belang van beoordelen wordt bevestigd door een survey dat in het kader van dit onderzoek is uitgevoerd (Punter en Lami, 1998). Op de vraag ‘wat zijn de redenen om een beoordeling uit te voeren?’ gaf het grootste deel van de (39) respondenten als antwoord: het bepalen van de softwareproductkwaliteit en het verbeteren van die kwaliteit. Andere redenen die werden genoemd zijn: selecteren van een softwarepakket, bereiken van acceptatie bij de klant en het reduceren van de onzekerheid in het ontwikkelingstraject.

1.2 Een softwareproduct beoordelen

In deze paragraaf wordt het begrip softwareproduct-beoordeling gepositioneerd. Dit gebeurt door het begrip eerst te definiëren (1.2.1), vervolgens te koppelen aan verwante begrippen (1.2.2) en een overzicht van beoordelingsmethoden te geven (1.2.3). Tenslotte worden twee relevante ISO-standaarden besproken (1.2.4).

1.2.1 Beoordelen is meten

Beoordelen wordt in de spreektaal omschreven als ‘een oordeel vellen, zijn goed- en of afkeuring uitspreken over’ (Geerts en Heestermans, 1984). Het Engelse equivalent van beoordelen is evaluation. Dit wordt door Webster’s (1996) gedefinieerd als: ‘to find or determine the amount, worth, etcetera, of’. De ISO 14598-1 (1999)- en de ISO 9126 (1999)-standaarden definiëren beoordelen van softwareproduct kwaliteit als: ‘systematic examination of the extent to which an entity is capable of fulfilling specified requirements’.

Kwaliteit wordt uitgedrukt door eigenschappen te specificeren waaraan het product moet voldoen. Zo is bijvoorbeeld gespecificeerd dat het product onderhoudbaar of bruikbaar moet zijn. Tijdens een beoordeling wordt de aanwezigheid van die eigenschappen bepaald. Meten speelt hierbij een belangrijke rol. Meten houdt in dat er waarden aan de eigenschappen worden toegekend. Met meten wordt bepaald of het product voldoet en zo ja, in hoeverre het voldoet aan de voor het product geldende eisen.

Dat er tijdens beoordelen wordt gemeten komt goed tot uiting in de Webster-definitie. Daar wordt immers gesproken van ‘determine the amount, worth’. Ook de ISO-definitie verwijst met het begrip ‘systematic examination’ naar meten. Meten is nodig om tot een systematisch oordeel te komen. De ISO-werkgroep 7 –die werkt aan de standaarden ISO 9126 en ISO 14598– geeft aan dat meten nodig is omdat een beoordeling objectief en reproduceerbaar moet zijn. Andere bronnen die in deze richting wijzen zijn (Willumeit, 1993) en (Welzel e.a., 1993). In hoofdstuk 2 gaan we hier verder op in. Op dit moment is het belangrijk om te onderkennen dat meten essentieel is voor het beoordelen van softwareproducten.

1.2.2 Verschillende termen voor beoordeling

Naast het begrip beoordeling zijn er andere begrippen in omloop. Testen, assessment en auditing zijn voorbeelden. In deze paragraaf wordt een overzicht van deze terminologie gegeven.

Het begrip *testen* wordt onder andere gebruikt door de standaarden die betrekking hebben op het certificeren –en daarmee beoordelen– van producten: de EN45001 (1989), EN45002 (1989) en EN45011 (1989). De beoordelende instantie wordt aangeduid als testing laboratory. Een test is 'a technical operation that consists of the determination of one or more characteristics of a given product, process or service according to a specified procedure’ (EN45001, 1989).

Testen wordt ook gebruikt als het gaat om het beoordelen van de functionaliteit van een informatiesysteem. In dit verband wordt het begrip acceptatietest gebruikt. Dit is echter niet sluitend. Zo definiëren Pol e.a. (1995) testen breder door het op te vatten als 'het proces van plannen, voorbereiden en meten dat tot doel heeft de kenmerken van informatiesystemen vast te stellen en het verschil tussen de actuele en de vereiste status aan te tonen'.

Verificatie en validatie zijn eveneens termen die in de context van beoordelen worden gebruikt. ISO 14598 (1999) en Bache and Bazzana (1994) gebruiken deze termen. Verificatie is ‘confirmation by examination and provision of objective evidence that specified requirements have been fulfilled’. Validatie betreft ‘confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled’.

Auditing is afkomstig uit de wereld van accountants en controllers, werkzaam bij bijvoorbeeld KPMG en PriceWaterhouseCoopers. Er zijn verschillende audit-vormen: EDP-auditing, system-audit, proces- en product-audit. Een audit is een formele beoordeling en resulteert normaliter in een formele (accountants)verklaring. ISO 8402 (1994) definieert (quality) audit als ‘systematic and independent examination to determine whether quality activities and related results comply with planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve objectives’. In de praktijk wordt er aan audits een zwaarder gewicht toegekend dan bijvoorbeeld aan een

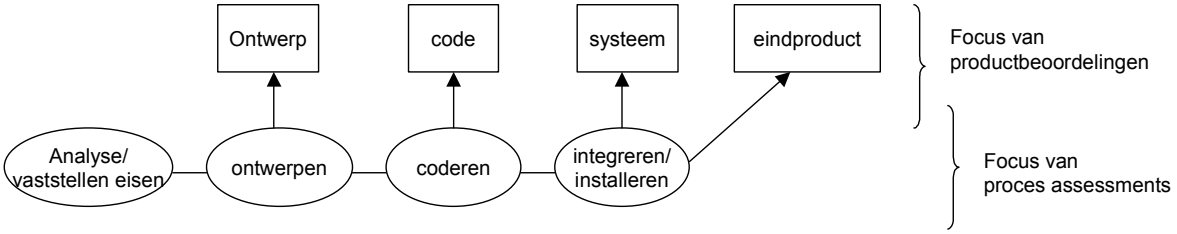
beoordeling. Informatie over auditing is te vinden in bijvoorbeeld (Praat en Suerink, 1992), (de Bruin, 1993), (Smith en Edge, 1991), (Moonen, 1991) en (Kocks, 1997).

Certificatie is een beoordeling waarbij een certificaat wordt afgegeven. Certificatie is, net als auditing, te beschouwen als een strenge en intensieve manier van beoordelen. Er wordt een certificatiesysteem gebruikt om tot een oordeel te komen. Dergelijke certificatiesystemen kunnen geaccrediteerd zijn. Met accreditatie wordt er vertrouwen –van de overheid, middels de Raad van Accreditatie– in het systeem uitgesproken. Het af te geven certificaat wordt gebruikt om het product toegang tot de markt te geven. Bekende certificaten zijn KemaKeur en CE. Voorbeelden van certificaten voor softwareproducten zijn: ICEK (1997), NF Logiciel (Geyres, 1997) en ScopeMark (O'Duffy e.a., 1999). Het certificeren van softwareproducten bevindt zich nog in een beginstadium: er zijn nog geen algemeen geaccepteerde certificaten.

Audit en certificatie worden ook wel aangeduid als *derde partij beoordeling*. Derde partij wil zeggen: een *onafhankelijke* partij naast de eerste partij (de producent, maker van de software) en tweede partij (klant/gebruiker, afnemer van de software).

Assessment is een begrip dat regelmatig valt in de context van softwareproductbeoordelingen. Voorbeelden zijn: architectural assessment (Kazman e.a., 1996) en conformity assessment (Lloyd's Register, 1994). In dit proefschrift wordt deze term gereserveerd voor software*proces*beoordelingen. Dit gebeurt conform Humphrey (1989) die assessment definieert als: 'a review of a software organisation to provide a clear and factual understanding of the organisation's state of software practice'. Voorbeelden zijn de Bootstrap- en CMM-assessments.

Met het begrip procesbeoordeling komt de relatie tussen product- en procesbeoordelingen aan de orde. Procesbeoordelingen richten zich op het ontwikkelingsproces zelf. Er wordt vastgesteld in hoeverre het proces voldoet aan de key process areas (kpa's) (Paulk e.a., 1995). Hierbij kan het resultaat van een proces (tussenproducten of zelfs het uiteindelijke softwareproduct) worden meegenomen. Bij productbeoordelingen is het net andersom. De focus ligt daar op de tussenresultaten of het eindproduct. Daarnaast wordt soms gekeken naar het ontwikkelingsproces. De onderstaande figuur beeldt deze verschillen in focus uit.



Figuur 1.2 Focus van proces assessment en productbeoordeling

De figuur laat zien dat focus van proces assessment op de ontwikkelingsactiviteiten ligt: analyse, ontwerpen, enzovoorts. De focus van productbeoordeling ligt op de resultaten van de activiteiten, zoals het ontwerp, de code et cetera. De figuur geeft ook aan dat er een overlap in focus bestaat. Tijdens een productbeoordeling kunnen aspecten van het proces worden meegenomen, terwijl tijdens een procesbeoordeling ook naar productaspecten wordt gekeken.

1.2.3 Beoordelingsmethoden

Softwareproductbeoordelingen worden over het algemeen uitgevoerd met behulp van een beoordelingsmethode (Eng.:evaluation method). In deze paragraaf geven we een overzicht van dergelijke methoden. Hiervoor karakteriseren we deze methoden naar de vier manieren die Bache en Bazzana (1994) onderkennen om een beoordeling uit te voeren, namelijk: inspecteren, statische analyse, dynamische analyse en modelleren. Deze indeling komt voort uit het Scope-project (Robert, 1994) en wordt vaak gebruikt om methoden te rubriceren. [2]

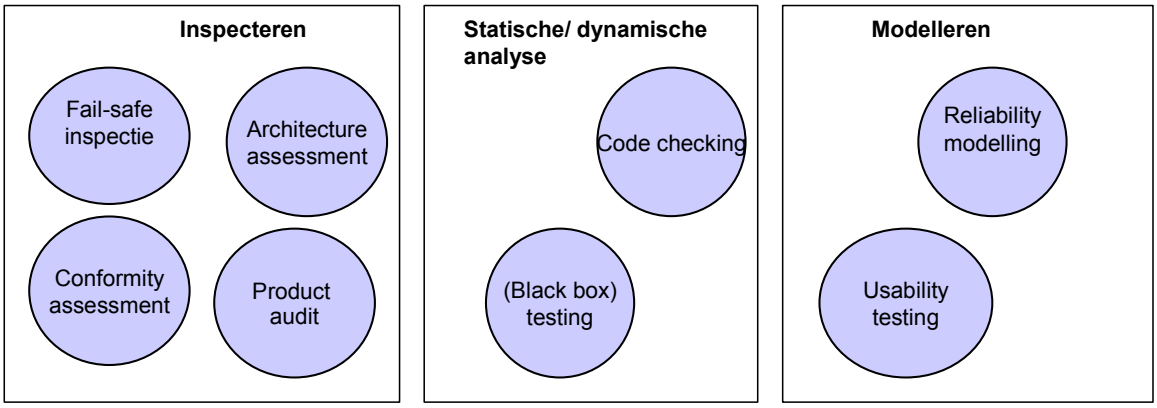
Het inspecteren betreft beoordelingen door personen. Bache en Bazzana (1994) geven aan dat de personen hierbij veelal zonder tool-ondersteuning de beoordeling uitvoeren. In dit proefschrift onderkennen we twee categorieën binnen inspecteren, namelijk: expert- en checklistgebaseerde beoordelingen. De expertbeoordeling wordt uitgevoerd door personen die verstand hebben van software in het algemeen en van het gebied waarop een concreet softwareproduct wordt toegepast. De checklistgebaseerde beoordeling wordt eveneens door personen uitgevoerd, maar deze worden daarbij ondersteund door een checklist of vragenlijst.

Statische analyse betreft de analyse van source code, bij dynamische analyse wordt de code geëxecuteerd. Dit soort analyses kan handmatig worden uitgevoerd, maar vaak gebeurt het

[2] Er zijn ook andere manieren om beoordelingsmethodes te karakteriseren. Voorbeelden zijn: a) op basis van het type software product waarover een uitspraak wordt gedaan: embedded software product, administratief informatiesysteem, b) het moment in de levenscyclus van het product waarop de beoordeling plaatsvindt: de architectuur van het product, de code of het kant-en-klare product én c) de scope van het product: code, documentatie of systeem

geautomatiseerd met speciaal daarvoor ontwikkelde compilers en parsers (Dumke e.a., 1996).

Modelleren betreft een methode waarin het beschrijven van de producteigenschappen in de vorm van een model centraal staat. Een beoordeling wordt dan uitgevoerd door een specifiek product te vergelijken met de eisen die in het model staan verwoord. Deze vergelijking gebeurt veelal geautomatiseerd. Voor elk van deze beoordelingsmethoden worden hieronder een aantal typerende gevallen genoemd, zie volgende figuur.



Figuur 1.3 Overzicht van beoordelingsmethoden

Een deelcategorie van inspecteren in figuur 3 betreft beoordelingen door experts. Het inspecteren van fail-safe systemen is een voorbeeld. Deze worden uitgevoerd door experts die bekend zijn met het bedrijfsproces in kwestie, bijvoorbeeld een chemisch proces. Zij kunnen daardoor de risico's inschatten van het implementeren van software in de besturing van een dergelijk proces. Op basis van deze kennis geeft de expert een oordeel over de software. Een voorbeeld is het beoordelen van de specificaties van de software voor de Maasland stormvloedkering in de Nieuwe Waterweg. De software betreft de functionaliteit voor het openen en sluiten van sluisdeuren. De expert doet een uitspraak over de betrouwbaarheid en de veiligheid van zo'n systeem.

Expertbeoordelingen worden als methode meestal gekozen bij eenmalige, grote projecten, zoals de bouw van een stormvloedkering, de invoering van Programmable Logic Controllers in een chemisch proces of de invoering van software-besturing in tram of trein (Friedman en Voas, 1995) en (Wijbrans e.a., 1999). Bij dergelijke projecten gaat het veelal om complexe maar vooral totaal nieuwe software.

Een andere vorm van expertbeoordeling betreft de beoordeling van een software architectuur. Dit wordt ook wel aangeduid met Architecture Analysis of Architecture Assessment. Over dergelijke beoordelingen is veel gepubliceerd door medewerkers van het Software Engineering Institute (SEI) van de Carnegy Mellon Universiteit in de Verenigde Staten (Abowd e.a., 1996), (Barbacci e.a., 1996). Dit heeft geleid tot een Software Architecture

Analysis Method (SAAM) (Kazman e.a., 1996). SAAM gaat ervan uit dat er scenario's moeten worden opgesteld voor een software architectuur. Dit houdt in dat er verschillende combinaties van software- en hardwarecomponenten worden geïdentificeerd en uitgewerkt. Een expert geeft dan een oordeel over de mogelijk optredende problemen bij de verschillende scenario's. SAAM is verder aangevuld in de methode ATAM (Kazman e.a., 1998). Er zijn plannen om ook te gaan meten aan architecturen (Briand e.a., 1998).

Een tweede categorie van inspecteren vormen de checklistgebaseerde beoordelingen. Bekende voorbeelden van dergelijke beoordelingen zijn conformity assessments en product audits. *Conformity assessment* betreft een beoordeling waarbij van een product wordt bepaald in hoeverre het voldoet aan een bepaalde set eigenschappen, bijvoorbeeld security of interoperability. Voorbeelden zijn: beoordelingen op basis van ISO 12119 (1994) waarbij wordt bepaald of een product voldoet aan de zes ISO 9126 kwaliteitskarakteristieken (Pivka, 1996), (Lloyd's Register, 1994). *Product audits* stellen allereerst vast welke karakteristieken van een product van belang zijn om vervolgens te beoordelen of en in welke mate voldaan wordt aan die karakteristieken. Methoden op dit gebied zijn: MicroScope (Kyster, 1995), Ibisco (Maiocchi, 1997), Q-Seal (Caliman, 1996) en TAQS (Tsukumo e.a., 1996).

De middelste kolom uit figuur 3 betreft dynamische en statische analyse. *Dynamische analyse* wordt ook wel aangeduid als testen. Een voorbeeld hiervan is het testen van systemen voor de gemeentelijke basisadministratie (GBA). Volgens het 'black box'-test principe wordt gecontroleerd of het berichtenverkeer van gemeenten en afnemers, over het GBA netwerk voldoet aan de in het logisch ontwerp gestelde eisen. Het verzenden van de testberichten en het vergelijken van ontvangen berichten met referentieberichten gebeurt geautomatiseerd.

Bij *statische analyse* wordt de source code beoordeeld. Er wordt bijvoorbeeld nagegaan of de code voldoet aan bepaalde ontwerprichtlijnen. Hiervoor wordt veelal een geautomatiseerd hulpmiddel –een zogenaamde code checker– gebruikt. Zo werden omstreeks 1975 op het Royal Signals and Radar Establishment in Malvern (UK) al programma's geanalyseerd. Men vond dat een programma van slechts 2 pagina's het meest complex was van alle geanalyseerde programma's in termen van 'number of edges' en 'number of nodes'. In de hoofdstukken 3, 4 en 7 komen beoordelingen gebaseerd op statische analyse aan de orde.

De rechter kolom uit figuur 3 betreft beoordelingen op basis van modellen waarin criteria over een softwareproduct zijn vastgelegd. Reliability growth modelling en de SUMI-methode zijn hier voorbeelden van. Bij *Reliability growth modelling* wordt een model opgesteld over het foutgedrag in software. Op basis van dit model wordt voorspelt wat de betrouwbaarheid (de kans van het optreden van nieuwe fouten) van de software is. Voor uitgebreide behandeling van deze benadering zie (Fenton en Pfleeger, 1996).

SUMI is een methode voor het beoordelen de van de Usability van softwarepakketten. De methode maakt gebruik van een model dat specificeert aan welke eisen een product dient te voldoen, wil het als Usable worden gekwalificeerd. Dit model is ontstaan op basis van verschillende beoordelingen in het verleden. SUMI hanteert een standaardvragenlijst. Na beantwoording van deze vragenlijst worden de resultaten hiervan ingevoerd in een algoritme dat tot een oordeel over het betreffende product komt (Kirakowski en Corbett, 1993)[3].

Tot zover een overzicht van de vele beoordelingsmethoden. We besluiten met de opmerking dat een beoordelingsmethode ook een combinatie van inspecteren, dynamische- en statische analyse en modelleren kan zijn. Zo is bij SUMI naast modelleren ook sprake van inspecteren, er wordt immers gestart met het inschakelen van personen voor de beantwoording van de vragenlijst.

1.2.4 Twee relevante ISO-standaarden

Dit proefschrift verwijst een aantal keren naar standaarden. Deze sub-paragraaf geeft een korte toelichting van de meest relevante standaarden voor het beoordelen van een softwareproduct.

De ISO 9126-standaard heeft als titel ‘Vocabulary to define software quality’. De standaard is in feite het woordenboek om software-kwaliteit te definiëren. De standaard geeft aan dat er zes karakteristieken zijn, verfijnd in 21 subkarakteristieken. De eerder besproken figuur 1 geeft de structuur van dit kwaliteitsmodel schematisch weer. De eerste officiële versie van deze standaard is in 1991 gepubliceerd. De laatste jaren is men bezig om tot een geactualiseerde versie te komen. Naast het feit dat de definities van een aantal (sub)karakteristieken is gewijzigd, bevat de nieuwe versie drie bijlagen. Elke bijlage bevat voorstellen voor metrieken. Men gaat in op respectievelijk externe, interne en quality-in-use metrieken. In hoofdstuk 2 komen we op dit onderscheid terug.

De ISO 14598-standaard heeft als titel ‘Evaluation process definition’. Deze standaard geeft aan welke deelprocessen en activiteiten er tijdens een (softwareproduct) beoordeling plaats moeten vinden. Ook geeft de standaard richtlijnen (do’s en do-nots) voor het uitwerken van de activiteiten. De standaard zelf bestaat uit zes delen:

- Part 1 General overview.
- Part 2 Planning and management.
- Part 3 Process for developers.
- Part 4 Process for acquirers.
- Part 5 Process for evaluators.

---

[3] Naast SUMI zijn inmiddels ook Mumms en Wammi als methodes voor usability beoordeling ontwikkeld. Mumms richt zich op multi media applicaties, Wammi op world wide web-applicaties.

– Part 6 Documentation of Evaluation Modules.

De standaard onderkent drie perspectieven van waaruit men een beoordelingsproces kan uitvoeren: het perspectief van ontwikkelaars, het perspectief van klanten en tot slot het perspectief van beoordelaars. In hoofdstuk 3 zullen we daarover uitweiden. Het ISO-14598 standaardisatieproces is gestart in 1992[4]. Er zijn 'commission drafts' (CD's) gepubliceerd in 1995 and 1996. Deze drafts zijn sterk beïnvloed door het Esprit II Scope project (Robert, 1994). Op dit moment zijn de delen 1, 4 en 5 officieel gepubliceerd.

1.3 Ontevredenheid over softwareproductbeoordelingen

Tijdens dit onderzoek is ervaring opgedaan met het ontwikkelen en uitvoeren van softwareproductbeoordelingen. Op basis van deze ervaringen wordt geconstateerd dat er in de praktijk ontevredenheid bestaat over deze beoordelingen. Om dit te illustreren, wordt de satisfactie over acht verschillende softwareproductbeoordelingen gepresenteerd. Het betreft beoordelingen van zeer diverse producten in de periode 1996 tot 1998. Het product betreft in alle gevallen software code en documentatie. Eén beoordeling (nummer 8) betrof het ontwerp van een systeem. Voor elk van de beoordelingen is door de onderzoeker een oordeel gevormd over de mate waarin men wel of niet tevreden was over een beoordeling, zowel qua resultaat als qua proces. Dit is gebeurd op basis van opmerkingen en opinies van zowel klant als beoordelaar tijdens en na de beoordelingen. De volgende tabel geeft de resultaten van deze inschatting.

Tabel 1.1 (On)tevredenheid over softwareproductbeoordelingen

	Beoordeling	Tevreden	Niet-tevreden	
			Resultaten niet gebruikt	Beoordeling stopgezet
1	Besturingssysteem voor energietoevoer	X		
2	Module in een embedded softwareproduct		X	X
3	Software spelletjes	X		
4	Hypotheek informatiesysteem		X	
5	Projectmanagement informatiesysteem	X		
6	Persoonsregistratie systeem		X	
7	Message handling systeem	X		
8	Software architectuur			X

De tabel laat zien dat de belanghebbenden bij een beoordeling in vier gevallen tevreden waren en in vier andere gevallen ontevreden. De ontevredenheid heeft betrekking op de effectiviteit van een beoordeling: aan de verwachting die de klant heeft wordt met het uitvoeren van de beoordeling niet voldaan. In één beoordeling (nummer 6) speelt naast effectiviteit ook mee dat de kosten van de beoordeling als te hoog werden ervaren.

[4] Zie voor het proces van standaardisatie: (Magee en Tripp, 1997) Rout (1998).

Ontevredenheid over beoordelingen leidt tot een tweetal reacties. De eerste soort reactie is dat de resultaten van de beoordeling niet worden gebruikt. De tweede soort is dat men zo ontevreden over de wijze van beoordelen is, dat men die beoordeling volledig aan de kant schuift.

Een andere indicatie over tevredenheid wordt afgeleid uit een survey dat tijdens dit promotieonderzoek is uitgevoerd (Punter en Lami, 1998). Hierbij zijn aan 31 Europese organisaties –alle gebruikers of producent van software– vragen over het beoordelen van software-kwaliteit gesteld. Gevraagd is of men tevreden is over de uitgevoerde beoordelingen van de software-kwaliteit. Van de 16 respondenten geven er 12 (75%) aan dat ze matig tevreden zijn over de uitgevoerde beoordeling. Men zou beoordelingen verbeterd willen zien.

Onderzoek naar het beoordelen van geïmplementeerde informatiesystemen door Miller en Dunn (1997) laat een vergelijkbaar beeld zien. Zij hebben 150 grote en kleine Engelse organisaties geënquêteerd. Er waren 33 respondenten. De vraag ‘welke problemen komt u tegen bij het beoordelen van informatiesystemen’ resulteerde in de volgende antwoorden:

- De resultaten worden nauwelijks gebruikt. Aangegeven door 20 respondenten (60%), 10 daarvan ervoeren dit als een groot probleem.
- De resultaten zijn onbetrouwbaar. Aangegeven door 19 respondenten (57%), waarvan 6 personen zeiden dit als een groot probleem te ervaren.
- Lessen uit een beoordeling worden nauwelijks toegepast. Aangegeven door 18 personen (54%), waarvan er 8 aangaven dit als een groot probleem te ervaren.

Deze cijfers demonstreren dat een groot aantal van de respondenten (55 tot 60 %) ontevreden is over de binnen hun organisatie uitgevoerde beoordelingen.

Een andere indicatie die op ontevredenheid over beoordelingen wijst, is dat er behoorlijk wat gevallen bekend zijn waar een aangeschaft tool voor het beoordelen van software-kwaliteit na verloop van tijd niet meer gebruikt wordt. Over dit niet meer gebruiken van meettools bestaan verschillende cijfers variërend tussen 25% en 40% (gehoord in de praktijk). Een veel gehoorde reden voor het terzijde schuiven is, dat de uitspraken die het tool over de kwaliteit van de code doet, niet overeenkomen met het beeld dat engineers en gebruikers over de software hebben.

Het bovenstaande laat zien dat men in de praktijk zeer vaak ontevreden is over softwareproduct- beoordelingen. Ontevredenheid manifesteert zich in met name in twee zaken:

- Een beoordelingsmethode wordt niet (meer) gebruikt: de ontwikkelde beoordeling wordt slechts één of enkele keren toegepast en dan aan de kant geschoven.

– De resultaten van een uitgevoerde beoordelingen worden niet gebruikt met als redenen: geen vertrouwen in het resultaat of het resultaat is niet bruikbaar en leidt niet tot verbeteringen van de software in kwestie.

1.4 Probleemstelling en onderzoeksdoel

De in voorgaande paragraaf geconstateerde ontevredenheid wijst erop dat er iets mis is met softwareproduct-beoordelingen. Dat is voor ons reden om na te gaan wat er nu precies schort aan beoordelingsmethoden en beoordelingsprocessen, te meer omdat, zoals eerder is aangegeven, software-kwaliteit een uiterst belangrijk onderwerp is.

Onze ervaring leert dat problemen met beoordelingen niet zo zeer liggen op het gebied van het vinden van geschikte metrieken of geautomatiseerde hulpmiddelen. Veeleer gaat het om de besturing van het beoordelingsproces. Er worden weliswaar diverse beoordelingsactiviteiten uitgevoerd, maar het ontbreekt stelselmatig aan een behoorlijke aansturing van activiteiten, het afstemmen tussen doel en middelen, laat staan dat men terugkoppelt om na te gaan of het gestelde doel is bereikt. Het voorgaande brengt ons op de probleemstelling van dit onderzoek, namelijk:

beoordelingsprocessen worden onvoldoende bestuurd.

Het gevolg hiervan is dat de betrokken partijen (opdrachtgever, ontwikkelaar, eindgebruiker) aan het eind van een beoordelingstraject in heel wat gevallen ontevreden zijn met het bereikte resultaat. Men herkent zich niet in de resultaten, weet niet hoe men geconstateerde lacunes in software-kwaliteit alsnog kan oplossen. Ook klaagt men over de gevolgde werkwijze en vindt men het resultaat wel erg pover lettende op het bedrag dat men heeft moeten investeren.

Kortom, er zijn veel klachten over beoordelingen, omdat niet wordt geleverd wat partijen verwachten. Daarbij wordt aangetekend dat men op zijn hoede moet zijn voor klakkeloos volgen van ontevredenheid over het beoordelingsresultaat. Er zijn immers verschillende partijen betrokken bij beoordeling die elk eigen belangen hebben. Stel, een leverancier van een softwarepakket moet onder druk van zijn potentiële afnemers, een kwaliteitsbeoordeling moet laten uitvoeren. Uiteraard is deze leverancier gebaat bij een goede uitkomst. Mocht nu de beoordeling als resultaat een onvoldoende opleveren, dan ligt het voor de hand dat de leverancier de beoordeling op allerlei punten zal aanvechten. De grens tussen terechte en onterechte kritiek is in dit geval moeilijk te trekken. Tevredenheid of ontevredenheid over beoordelingen moet men dus met de nodige voorzichtigheid hanteren.

Het feit dat partijen in veel gevallen terechte kritiek op beoordelingen kunnen hebben, is voor een belangrijk deel te verklaren uit het ontbreken van een algemeen geaccepteerd conceptueel kader voor beoordelingsprocessen en met name voor de aansturing van

dergelijke processen. Op dit punt schort het aan goede theorie. Dat brengt ons tot het wetenschappelijke probleem van dit proefschrift, namelijk:

> *het ontbreekt aan een goed onderbouwd conceptueel kader voor beoordelingsprocessen en voor met name de aansturing van die processen.*

Wat is nu het onderzoeksdoel van dit proefschrift? Gegeven het probleem in de praktijk en het geschetste wetenschappelijke probleem, willen we met dit onderzoek een analysekader opstellen om beoordelingsprocessen gestructureerd te kunnen beschrijven en aan te geven op welke punten welke besturing nodig zou zijn. Vervolgens leiden we uit dit analysekader een ontwerp af voor een verbeterde sturing van softwarebeoordeling.

Het gestelde doel leidt tot de volgende onderzoeksvragen:

- Wat is een beoordelingsproces, uit welke activiteiten bestaat zo'n proces, welke hulpmiddelen en metrieken worden daarbij gebruikt en welke effecten heeft dit?
- Wat is de besturing van een dergelijk proces, welke stuurmaatregelen kunnen door wie op welk moment genomen worden?
- Welke problemen doen zich in de praktijk voor bij de besturing van softwarebeoordelingen en wat zijn de nadelige gevolgen daarvan?
- Wat zijn mogelijke verbeteringen in het aansturen van beoordelingsprocessen en hoe kan men deze vastleggen in een ontwerp van een goed onderbouwd conceptueel kader?

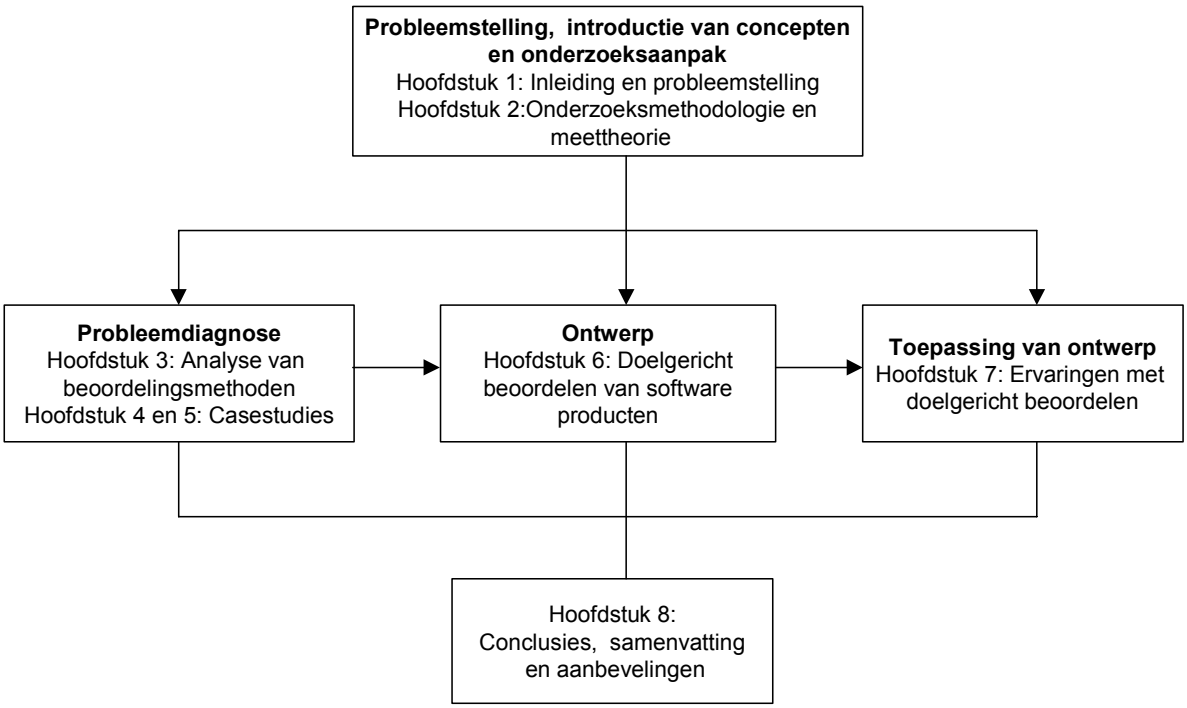
Deze onderzoeksvragen komen in de volgende hoofdstukken uitvoerig aan de orde.

1.5 Structuur van het proefschrift

Dit hoofdstuk had tot doel de probleemstelling van softwarebeoordelingen duidelijk te maken en een aantal relevante begrippen in dat kader toe te lichten. Hoofdstuk 2 gaat in op de methodologie van het onderzoek. Daarnaast wordt in dat hoofdstuk het belang van meten voor het beoordelen van softwareproducten toegelicht en wordt het begrip metrieken uitvoerig behandeld.

Hoofdstuk 3 definieert een analysekader voor beoordelingsprocessen. Vervolgens worden hiermee bestaande beoordelingsmethoden geanalyseerd. Dit resulteert in een meer uitgewerkte probleemstelling. In hoofdstuk 4 en 5 wordt deze probleemstelling getoetst aan de hand van twee casestudies.

Hoofdstuk 6 presenteert een ontwerp waarmee beoordelingsprocessen en de besturing daarvan verbeterd kunnen worden. Met dit ontwerp is in hoofdstuk 7 ervaring opgedaan in de praktijk. Hoofdstuk 8 eindigt met conclusies en aanbevelingen. Onderstaande figuur geeft de structuur van het proefschrift.



Figuur 1.4 Structuur van het proefschrift

2. Onderzoeksmethodologie en meettheorie

Dit hoofdstuk gaat in op zowel de methodologie die tijdens het onderzoek is gevolgd als de meettheorie die nodig is voor een beter begrip van softwareproduct-beoordelingen.

2.1 Onderzoeksmethodologie

De methodologie van het onderzoek wordt behandeld door eerst de aard van het onderzoek te beschrijven (2.1.1.). Vervolgens wordt ingegaan op de onderzoeksaanpak en de gevolgde strategie (2.1.2). Tenslotte worden de veronderstellingen geëxpliciteerd die ten grondslag liggen aan het onderzoek (2.1.3).

2.1.1 Aard van het onderzoek

Bedrijfskunde en software engineering vormen beide het domein van dit onderzoek. Er wordt vanuit bedrijfskundige principes naar het onderwerp van onderzoek –softwareproduct-beoordelingen– gekeken. Het onderwerp is tevens gerelateerd aan de software engineering (Shaw, 1990). Veel van de terminologie die in het proefschrift gebruikt wordt, is afkomstig uit deze discipline. Omdat het onderzoek zich richt op het beoordelen van software, rekenen we het tot het kennisgebied software management (Reifer, 1993), (Heemstra, 1994). Het onderzoek zelf kwalificeren we als ontwerpgericht en exploratief.

Het onderzoek is *ontwerpgericht* omdat het resulteert in een ontwerp. Hierbij wordt de ontwerpgerichte onderzoeksaanpak gevolgd (Van den Kroonenberg, 1974), (Florusse en Wouters, 1991). Dit houdt in dat na een grondige analyse problemen rondom het beoordelen van softwareproducten worden geïdentificeerd. Vervolgens wordt er aangegeven waarom de huidige manier van beoordelen niet voldoet, waarna een ontwerp ter verbetering wordt opgesteld. Ontwerpgericht onderzoek is normaliter praktijkgericht. Ook dit onderzoek houdt zich bezig met praktijkproblemen. Het is echter ook theoriegericht[5], er worden namelijk twee conceptuele modellen ontwikkeld: een analysekader om naar beoordelingen te kijken en een ontwerp ter besturing van beoordelingsprocessen.

Het onderzoek is *exploratief* omdat er veel aandacht wordt besteed aan het definiëren van concepten. Zo is besturingstheorie gebruikt om een analysekader op te stellen en komt het ontwerp via logisch redeneren tot stand. Het exploratieve karakter komt ook tot uiting in de nadruk op de onderzoeksaanpak, zie ook hieronder. Er is veel aandacht besteed aan de probleemstelling, het diagnosticeren ervan en het opstellen van het ontwerp.

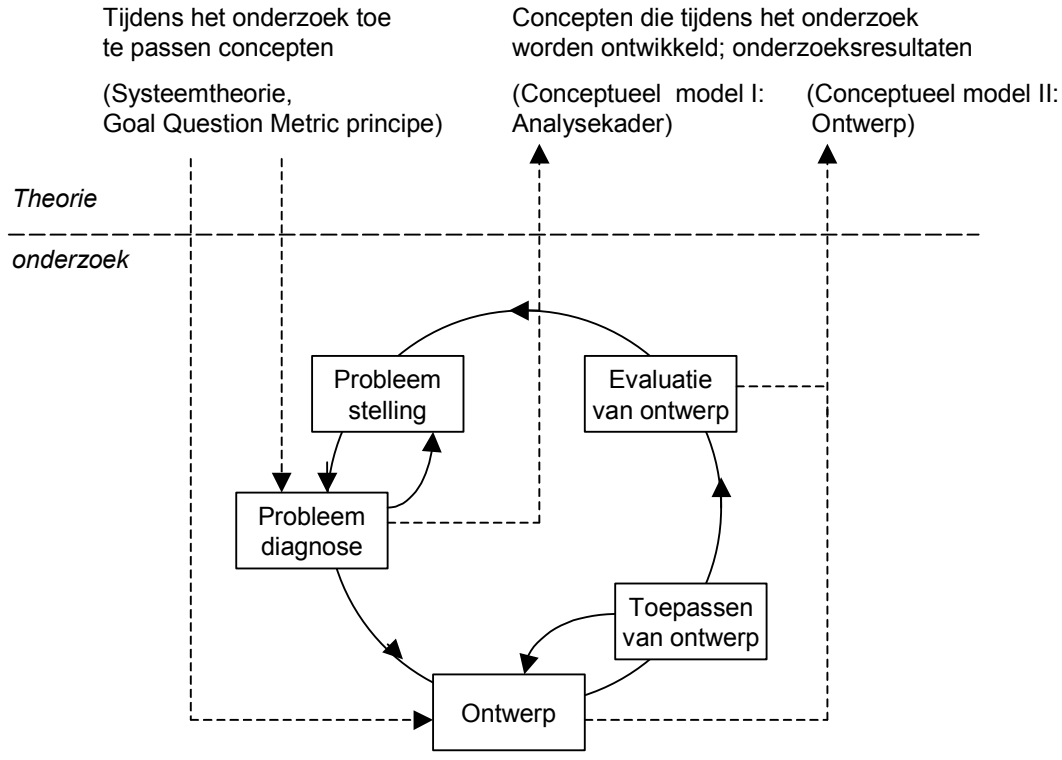
---

[5] Dit verwijst naar de indeling praktijk- versus theoriegericht onderzoek die Van der Zwaan (1995) en Verschuren en Doorewaard (1995) aanhouden.

2.1.2 Onderzoeksaanpak en -strategie

Deze subparagraaf beschrijft de onderzoeksaanpak en de gevolgde strategie tijdens het onderzoek. De aanpak betreft de fasen of stappen die tijdens het onderzoeksproces zijn genomen. De onderzoeksstrategie betreft de wijze waarop kennis in het onderzoek tot stand is gekomen. Het betreft 'het geheel van met elkaar samenhangende beslissingen over de wijze waarop het onderzoek wordt uitgevoerd'. Bij de uitvoering wordt met name gedoeld op het vergaren van relevant materiaal en de verwerking van dit materiaal tot antwoorden op de onderzoeksvragen' (Verschuren en Doorewaard, 1995). Voorbeelden van strategieën zijn: survey, experiment, gevalstudie en actie-onderzoek (Yin, 1994), Van der Zwaan (1995).

Voor de onderzoeksaanpak hebben we ons gebaseerd op (Van Aken, 1994) en (Van Eijnatten, 1992). Zij geven aan dat voor ontwerpgericht onderzoek zowel de regulatieve cyclus (Van Strien, 1986) als de reflectieve cyclus (van Aken, 1994) van belang zijn. De regulatieve cyclus is nodig om op basis van geconstateerde problemen in te grijpen in de praktijk. De reflectieve cyclus dient om na te gaan of de ingreep het gewenste effect heeft gehad. Van Eijnatten (1992) geeft een grondfiguur voor ontwerpgericht onderzoek, die er voor dit onderzoek als volgt uitziet:



Figuur 2.1 Fasen van de onderzoeksaanpak, afgeleid uit (Van Eijnatten, 1992)

Probleemstelling en diagnose

Het onderzoek is gestart met het definiëren van de probleemstelling. Deze is gebaseerd op het 'gevoel van onbehagen' en ontevredenheid over beoordelingen in de praktijk. In de

literatuur wordt hierop vrijwel niet ingegaan, vandaar dat dit onderdeel van het onderzoek gebaseerd is op praktijkervaringen. Parallel daaraan is een survey uitgevoerd om te bepalen welke factoren leidden tot ontevredenheid bij softwarebeoordelingen. Hiermee kwamen we tot het inzicht dat er bij dergelijke beoordelingen slecht gestuurd en beheerst wordt. Dit resulteerde in de probleemstelling, zoals beschreven in hoofdstuk 1.

Vervolgens is er een analysekader opgesteld om te definiëren wat een beoordelingproces nu precies is en wat er onder besturing van het proces moet worden verstaan. Hiervoor is literatuurstudie uitgevoerd: systeemtheorie is gebruikt om het analysekader te ontwikkelen. Dit kader vormt als het ware ‘de bril’ waarmee naar het onderzoeksonderwerp wordt gekeken.

Het analysekader is toegepast op in de literatuur beschreven beoordelingsmethoden. Dit leidde tot een meer aangescherpte probleemdiagnose in hoofdstuk 3 waarbij de bestaande manier van beoordelen is bekeken. Deze diagnose is getoetst middels casestudies (zie hoofdstukken 4 en 5). Tijdens het uitvoeren van deze casestudies stond de vraag centraal of de geformuleerde probleemstelling werkelijk overeenkomt met de problemen in de beoordelingspraktijk.

Strategie

De gevolgde strategie tijdens de hiervoor beschreven aanpak is een combinatie van logisch redeneren, van surveys, van literatuurstudie en van casestudies. Het logisch redeneren leidde tot de vaststelling van het probleem dat vervolgens is uitgewerkt in een diagnose. Surveys (Punter en Lami, 1998) speelden een rol bij het bepalen van de probleemstelling. Verder is hierbij literatuur over de systeemtheorie en zijn er ISO-standaarden gebruikt. Literatuurstudie was ook een belangrijke strategie waar het ging om de aanscherping van de diagnose. Het bestuderen van in de literatuur bekende beoordelingsmethoden leidde tot een gedetailleerde diagnose van het probleem (zie hoofdstuk 3). Vervolgens is deze probleemdiagnose bevestigd middels het uitvoeren van casestudies (zie hoofdstukken 4 en 5). Op deze laatste strategie gaan we hieronder verder in.

Casestudies

Een casestudie of gevalstudie is een onderzoeksstrategie waarbij er aan de hand van een protocol een geval in de praktijk wordt onderzocht. Yin (1994) geeft aan dat casestudies met name geschikt zijn voor onderzoeksvragen die zijn te kwalificeren als ‘hoe’ en ‘waarom’ vragen. Hij definieert een casestudie als ‘an empirical enquiry that investigates a contemporary phenomenon within its real-life context, when the boundaries between phenomenon and context are not evident and in which multiple sources of evidence are used’.

In bedrijfskundig onderzoek worden casestudies vaak toegepast. Van der Zwaan (1998) en andere auteurs (Yin, 1994), (Van Aken, 1994) wijzen op twee gangbare opvattingen over casestudies, namelijk:

1. Een gevalstudie is zeer geschikt om inzicht te krijgen in zowel het bestudeerde geval als in soortgelijke gevallen.
2. Een enkele gevalstudie leent zich niet voor theorievorming.

Van der Zwaan (1998) geeft aan dat deze opvattingen tevens misvattingen kunnen zijn. Wat betreft de eerste opvatting is het belangrijk dat men zich realiseert dat inzicht pas ontstaat door te vergelijken. Een losstaande casestudie levert nauwelijks inzicht op. In dit verband moet de uitspraak ‘één casus is geen casus’ worden geplaatst. Casestudies leiden alleen tot inzicht als de bevindingen worden vergeleken met de resultaten van andere casestudies of met bestaande theorieën.

De tweede opvatting betreft de generalisatie van de resultaten ofwel externe validiteit[6]. De opvatting gaat er van uit dat het bij casestudies om specifieke gevallen gaat en dat de ervaringen contextspecifiek zijn. Dit contextspecifieke maakt dat er niet kan worden gegeneraliseerd. Van der Zwaan (1998) geeft aan dat dit een misvatting is. Immers, als een ontwerp wordt toegepast middels een aantal casestudies en het blijkt dat het ontwerp in die gevallen werkt, dan leiden deze ervaringen ook tot een theorievorming. Problematischer is het om vervolgens op basis van deze theorie tot algemene uitspraken te komen. Dit inductieprobleem speelt ook bij andere onderzoeksstrategieën. Het wordt veelal ondervangen door de theorie meerdere keren toe te passen. Het aantal keren dat een theorie wordt bevestigd is dan van belang. Naarmate men vaker witte zwanen tegenkomt, is de uitspraak ‘alle zwanen zijn wit’ sterker. Op het moment dat men echter een zwarte zwaan tegenkomt wordt de uitspraak gefalsificeerd (Popper, 1968).

Het herhaalt toepassen van casestudies is moeilijk. Allereerst omdat het moeilijk is om een uitgevoerde casus onder precies dezelfde omstandigheden te herhalen. Er zullen altijd andere (omgevings-)invloeden zijn. Daarnaast kost het uitvoeren van casestudies veel tijd en inspanning. Yin (1994) en Van der Zwaan (1998) geven daarom het advies om de gevallen zodanig te selecteren dat ze betrekking hebben op het gehele domein waarover men een uitspraak wil doen.

Dit is de reden waarom de casestudies in dit onderzoek zo gekozen zijn dat er twee uiteenlopende benaderingen op het gebied van softwareproduct-beoordelingen worden afgedekt, namelijk: de op codemetrieken gebaseerde en de vraaggebaseerde beoordeling. In

---

[6] Externe validiteit heeft te maken met de kwaliteit van de reikwijdte van een conclusie uit een onderzoek: in hoeverre is de conclusie generaliseerbaar naar andere situaties? Een verwant begrip is interne validiteit. Dit is kwaliteit van de conclusie uit een onderzoek: is de interpretatie van de in de realiteit geconstateerde samenhang tussen verschijnselen juist? (Hutjes en Van Buuren, 1996).

hoofdstuk 3 wordt uitgewerkt wat we hieronder verstaan. De cases representeren ieder een beoordelingsbenadering. Zo proberen we enigszins te generaliseren. Dit is nodig om met de casestudies aan te tonen dat de gesignaleerde problemen rondom de besturing van beoordelingsprocessen een wezenlijke problematiek is.

Ontwerp, toepassing en evaluatie

Na de probleemanalyse is een ontwerp opgesteld. Dit is gedaan door middel van logisch redeneren en literatuurstudie. Hierbij zijn de probleemgebieden, die tijdens de analyse zijn opgesteld, het uitgangspunt. De richtlijnen zijn voor deze gebieden ontwikkeld. Dit leidde niet in één keer tot de volledige set richtlijnen, zoals de set opgesteld in hoofdstuk 6. Er waren verschillende iteraties nodig om deze set te ontwikkelen. Een aantal richtlijnen werd al toegepast tijdens praktijkopdrachten, terwijl andere richtlijnen nog moesten worden ontwikkeld.

Het ontwerp is toegepast tijdens drie opdrachten in de praktijk. In elk van de opdrachten betrof het een organsiatie die behoefte had aan een advies over het beoordelen van software. Tijdens de opdrachten zijn niet alle richtlijnen toegepast. Het finale ontwerp is pas laat in het onderzoekstraject opgesteld; er is sprake van exploratief onderzoek, waardoor niet alle richtlijnen toegepast konden worden.

Het ontwerp is geëvalueerd door drie evaluatieniveaus te onderkennen, namelijk: logische consistentie, toepasbaarheid en tevredenheid. Het eerste niveau komt in hoofdstuk 6 aan de orde als de rationale achter het ontwerp wordt beschreven. De andere twee evaluatieniveaus worden in hoofdstuk 7 ingevuld.

Om de toepasbaarheid van het ontwerp te evalueren zijn een aantal richtlijnen in de praktijk uitgevoerd. De werking ervan kon zo worden geëvalueerd. Van de richtlijnen die niet actief zijn toegepast kon achteraf voor een deel worden nagegaan of de acties die met de richtlijn beoogd werden, überhaupt voorkomen in de praktijk. Met deze manier van evalueren hebben we ervaring opgedaan over het ontwerp.

De evaluatie van de tevredenheid over het ontwerp is uitgevoerd door bij de beoordeling betrokken personen te interviewen omtrent hun ervaringen met de toegepaste richtlijnen.

De generaliseerbaarheid van de kennis over het ontwerp die is opgedaan tijdens deze opdrachten is beperkt. Door de opdrachten zo te kiezen dat beide beoordelingsbenaderingen –zowel de op codemetriek als de vraaggebaseerde benadering- aan bod komen proberen we aannemelijk te maken dat het ontwerp bruikbaar is in voor andere praktijkgevallen.

2.1.3 Veronderstellingen

Tijdens het uitvoeren van het onderzoek zijn een aantal veronderstellingen gehanteerd[7]. In deze paragraaf worden deze geëxpliciteerd. Het betreft:

- *Softwareproductbeoordelingen zijn maatwerk* Anders geformuleerd: er is niet één algemeen geldige manier om elk type softwareproduct te beoordelen. In de systeemontwikkeling wordt hiervoor de uitdrukking gebruikt: 'dé methode bestaat niet' (Van Rees, 1982). Een dergelijke manier om naar processen te kijken wordt bestempeld als contingentie- of situationele benadering. Dit onderzoek gaat uit van zo'n benadering omdat reeds in het begin werd geconstateerd dat softwareproducten op verschillende manieren worden beoordeeld. Een indicatie hiervoor is ook de diversiteit in beoordelingsmethoden die in paragraaf 1.2.3 aan de orde gekomen. De veronderstelling leidt ertoe dat er verschillende situationele factoren worden onderkend die het beoordelingsproces beïnvloeden. Dit is een reden om daaraan aandacht te besteden. Het heeft onder andere geleid tot een overzicht van de dimensies (Punter en Lami, 1998).
- *Relatie beoordelingsproces en ontevredenheid* – Er is een relatie verondersteld tussen de in hoofdstuk 1 geconstateerde ontevredenheid ten aanzien van beoordelingen enerzijds en het beoordelingsproces zelf anderzijds. De geconstateerde ontevredenheid wordt als uitgangspunt genomen om naar het beoordelingsproces te kijken. Daarbij wordt duidelijk gemaakt dat de ontevredenheid terecht is en voortkomt uit problemen met –de besturing– van het beoordelingsproces. Vervolgens is geprobeerd hiervoor verbeteringen voor te stellen,
- *Beoordelen betekent meten* – Meten neemt een centrale plaats in tijdens het beoordelen van softwareproducten. Het beoordelingsproces wordt dan ook vanuit een meetperspectief ingericht,
- *Beoordelen betekent afwegen van doel en middelen* – een beoordelingsproces wordt uitgevoerd om een doel te bereiken, hiervoor zijn middelen nodig. Deze bedrijfskundige invalshoek leidt tot een nadere bestudering van algemene principes uit de besturingstheorie om vervolgens na te gaan op welke wijze het beoordelingsproces bestuurd moet worden en waar verbeteringen nodig zijn.

2.1.4 Samenvatting

Dit onderzoek wordt getypeerd als ontwerpgericht en exploratief onderzoek. Het onderzoek resulteert in twee conceptuele modellen: een analysekader en een ontwerp ter verbetering van de bestaande beoordelingspraktijk. Elk van deze modellen vereist zijn eigen onderzoeksstrategie. In het eerste geval leunt deze strategie zwaar op casestudies. In het

---

[7] Methodologen maken nog een onderscheid tussen veronderstellingen en zogenaamde sensitising concepts. Het laatste zijn dan theoriegeladen noties die de onderzoeker impliciet of expliciet hanteert wanneer de verschijnselen in een casestudie worden bestudeerd. Een sensitizing concept is als het ware een niet-geëxpliciteerde veronderstelling.

tweede geval heeft het exploratieve karakter de overhand: er is door middel van logisch redeneren een ontwerp opgesteld, dat voor een deel in de praktijk is toegepast.

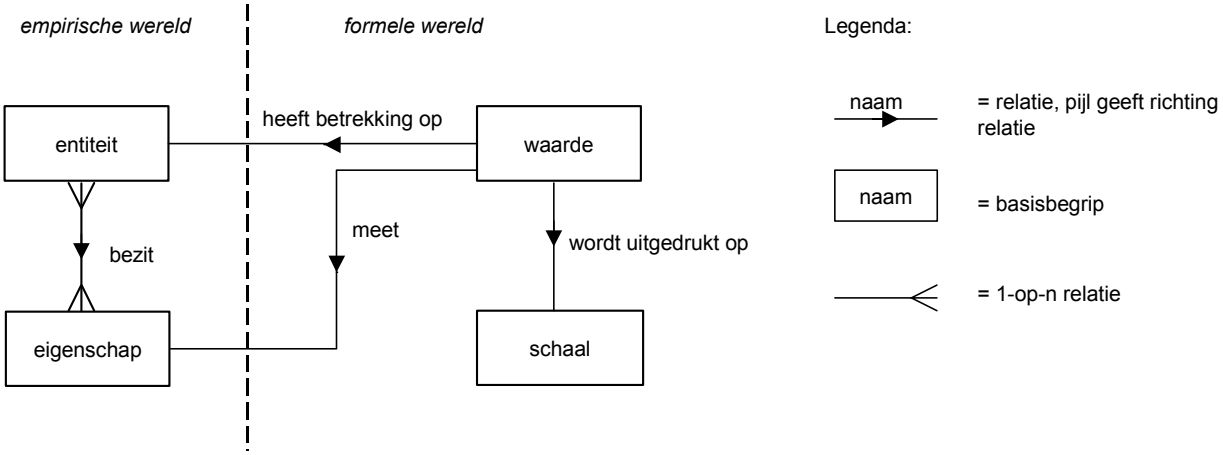
2.2 Meettheorie

In hoofdstuk 1 is aangegeven dat het beoordelen van softwareproducten moet worden uitgevoerd door te meten aan de software. In deze paragraaf wordt uitgewerkt wat er in dit proefschrift onder meten wordt verstaan.

2.2.1 Basisbegrippen

Verschillende auteurs definiëren *meten* als het proces waarin getallen of symbolen aan software-entiteiten worden toegekend om een eigenschap van deze entiteiten te karakteriseren. De toekenning van eigenschappen gebeurt volgens (meet)regels (Fenton en Pfleeger, 1996), (ISO 9126, 1991), (ISO 14598, 1999) en (Zuse, 1998). De getallen of symbolen zijn (meet)waarden. Deze worden uitgedrukt op een (meet)schaal. De schaal bepaald welke wiskundige bewerkingen –zoals delen of vermenigvuldigen– op de waarde mogen worden toegepast. De toegekende waarde en schaal rekenen we tot de zogenaamde formele wereld (Kitchenham e.a., 1995a). Dit is een beschrijvingsruimte waarin wiskundige bewerkingen worden uitgevoerd. Deze formele wereld is een afbeelding van wat er in de empirische wereld wordt waargenomen. Dit is de realiteit waartoe de software-entiteit en de eigenschappen ervan behoren. Door te meten beelden we de werkelijkheid af in waarden waarop wiskundige bewerkingen mogelijk zijn.

Met de definitie van meten zijn vier basisbegrippen geïntroduceerd, namelijk: entiteit, eigenschap, waarde en schaal. De begrippen en hun onderlinge samenhang worden in de volgende figuur gegeven. Deze figuur is afgeleid uit het ‘structural model for measurement’ zoals gedefinieerd door Kitchenham e.a. (1995a)



Figuur 2.2 Basisbegrippen om meten te beschrijven

Hieronder worden de begrippen uitgewerkt. Hiervoor wordt gebruik gemaakt van (Serc, 1992) en (Kitchenham e.a., 1995a).

Een entiteit is een object in de empirische wereld. In dit proefschrift wordt een *software entiteit* gedefinieerd als een onderdeel van het onderwerp van beoordeling. Voorbeelden zijn de specificaties van het systeem en de broncode. Er zijn verschillende soorten software entiteiten. Het kan gaan om een (onderdeel van) het product, een (deel van een) proces of om de hulpmiddelen die tijdens het proces worden gebruikt. We classificeren software entiteiten daarmee volgens de indeling product, proces en hulpmiddel (Fenton en Pfleeger, 1996). Onderstaande tabel geeft voorbeelden van entiteiten voor elk van de drie categorieën.

Tabel 2.1 Voorbeelden bij indeling product, proces en hulpmiddelen

Product	Proces	Hulpmiddel
Specificaties	Opstellen van specificaties	Personeel
Ontwerp	Opstellen ontwerp	Teams
Code	Coderen	Software
Test gegevens	Testen	Hardware

De metingen die tijdens het beoordelen van softwareproducten worden uitgevoerd zijn in eerste instantie gericht op de categorie product. In dit verband is de opmerking over de focus van productbeoordelingen in hoofdstuk 1 van belang. Er is daar aangegeven dat de focus weliswaar op product ligt, maar er kan ook worden gekeken naar (ontwikkel) activiteiten. De metingen tijdens productbeoordelingen zijn dan ook niet beperkt tot de categorie product. Ook entiteiten uit de andere categorieën kunnen tijdens deze beoordelingen aan de orde komen.

Elke entiteit bezit één of meer *eigenschappen*. Voorbeelden van eigenschappen van software entiteiten zijn de lengte van een module, de taal waarin de module is geprogrammeerd of het aantal fouten in de module. Figuur 2.2 geeft de relatie tussen entiteit en eigenschap. Een entiteit kan meer dan één eigenschap bezitten, terwijl een eigenschap gerelateerd is aan meer dan één entiteit.

Synoniemen voor het begrip eigenschap zijn attribuut (Eng.: attribute) en kwaliteitskarakteristiek (Eng.: quality characteristic). ISO 9126 kwaliteitskarakteristieken, zoals functionaliteit en onderhoudbaarheid, zijn daarmee eigenschappen. Hierbij is het belangrijk om een onderscheid te maken tussen de eigenschappen waaraan het product moet voldoen (de norm) en de eigenschappen die het product feitelijk bezit. Zo is bijvoorbeeld gesteld dat het product onderhoudbaar moet zijn. De meting aan het product moet dan aantonen of het product onderhoudbaar is. Hiervoor moeten normwaarden (synoniem: kwalificatieniveaus, grenswaarden) worden gedefinieerd. De actuele waarden die de uitvoer zijn van de meting moeten tussen deze normwaarden liggen wil het product voldoen aan de gestelde norm.

De waarden worden in de meetdefinitie hierboven aangeduid als getallen of symbolen. Met getal en symbool komen we op het onderscheid tussen kwantitatieve of numerieke eenheden (Eng.: numerical units) en kwalitatieve eenheden of kwalificaties. Een voorbeeld van een kwantitatieve eenheid is het getal 19 dat de waarde is van de eigenschap 'aantal regels code'. Een voorbeeld van een kwalificatie is de constatering dat de module is geïnspecteerd is (of niet).

Welke waarde kwantitatief is en welke kwalitatief hangt af van de schaal van de meting. *Schaal* en *eenheid* zijn synonieme begrippen. Zo is graad Celsius een eenheid op de Celsius schaal. Vaak is er meer dan één schaal om een waarde toe te kennen aan een eigenschap. Bekende voorbeelden hiervan zijn de schalen voor het meten van temperatuur: Kelvin, Celsius en Fahrenheit waarbij een bepaalde waarde op elke schaal een verschillende betekenis heeft. Dit gaat soms ook op voor het meten aan software: de waarde 10 op de schaal van het aantal regels code of op de schaal functiepunten hebben verschillende betekenis.

Het onderscheid tussen kwantitatieve en kwalitatieve eenheid is terug te voeren op het onderscheid in schaaltypen. We onderkennen vier s*chaaltypen*, namelijk: nominaal, ordinaal, interval en ratioschaal. Metingen waarbij de nominale of ordinale schaal wordt toegepast leiden tot wat we kwalitatieve metingen noemen. De meetwaarden zijn kwalificaties. De interval en ratio schaal worden toegepast tijdens wat we kwantitatieve metingen noemen. Deze resulteren in nummers, getallen. De volgende tabel geeft van elk schaaltype een omschrijving en voorbeeld.

Tabel 2.2 Vier schaaltypen (Serc, 1992).

Schaal type	Omschrijving	Voorbeelden
Nominale schaal	Op deze schaal is geen classificatie of rangschikking mogelijk. De meting bestaat uit het bepalen tot welke klasse een waarde hoort.	Het classificeren van de oorzaak van fouten als: specificatiefout, ontwerpfout, of code fout
Ordinale schaal	Op deze schaal is het mogelijk om naast classificeren ook te rangschikken. De waarde wordt gekozen uit een verzameling geordende categorieën.	Het classificeren van fouten als: fataal, major en minor
Interval schaal	Op deze schaal is het mogelijk om naast classificeren en rangschikken ook de afstand tussen twee waarden te bepalen. Dit betekent dat we kunnen zeggen dat de ene waarde *n* meer is dan een andere waarde. Het nulpunt is arbitrair gekozen.	Het scoren van intelligentie (IQ). De waarde 150 betekent 50 eenheden meer dan gemiddelde intelligentie
Ratio schaal	Dit is een interval schaal met een absoluut of natuurlijk nulpunt. Ook het bepalen van verhoudingen is zinvol.	Het aantal regels code, het aantal schermen op een afdeling.

Bij het beoordelen van softwareproducten hebben we te maken met zowel kwantitatieve als kwalitatieve metingen. Inspecties waarbij checklists worden gebruikt resulteren veelal in kwalitatieve uitspraken over het product. Een voorbeeld is de uitspraak over de leesbaarheid

van de documentatie, die als goed, gemiddeld of slecht wordt gekwalificeerd. We hebben te maken met een meting op ordinale schaal. Bij beoordeling met codemetrieken is daarentegen sprake van kwantitatieve metingen. De waarde 10 GoTo statements wordt uitgedrukt op een interval schaal. Het is geen ratioschaal omdat er moeilijk een absoluut nulpunt is aan te wijzen.

Nu de basisbegrippen zijn uitgewerkt komen we op het begrip metriek. De definitie van 'meten' geeft immers aan dat de toekenning van eigenschappen volgens (meet)regels dient te gebeuren. Er zijn meetvoorschriften nodig. De combinatie van meetvoorschrift én de schaal om de waarde van een eigenschap te bepalen is de metriek (ISO 9126, 1991), (ISO 14598-1, 1999). Metriek betreft dus zowel de schaal als het meetvoorschrift. Schaal is hiervoor aan de orde gekomen. Hieronder wordt op meetvoorschrift ingegaan.

Het meetvoorschrift beschrijft hoe de waarde voor een eigenschap wordt bepaald. Het volgende is een voorbeeld van een meetvoorschrift, 'Mean Time Between Failures' (Serc, 1992). Het luidt:

- bepaal welke storingen relevant zijn,
- bepaal hoe vaak of hoe lang er gemeten moet worden om een voldoende betrouwbare meting te verkrijgen,
- meet gedurende dit aantal keren of deze periode de tijd van opstarten van het softwareproduct tot het optreden van een storing,
- neem van deze tijden het gemiddelde.

Dit meetvoorschrift bepaalt de waarde op een ratioschaal. De schaal en het meetvoorschrift tezamen duiden we aan als de metriek 'Mean Time Between Failures'. Deze metriek kan worden gebruikt voor het bepalen van de eigenschappen 'bedrijfszekerheid' en 'beschikbaarheid' (Serc, 1992).

Een metriek is te beschouwen als een systeem met een invoer en een uitvoer. De uitvoer van de metriek is de gemeten waarde. De invoer bestaat uit de *(meet)gegevens* (Eng.: data) (Basili en Weiss, 1984). Een voorbeeld zijn 'de storingen' uit het voornoemde meetvoorschrift. De gegevens kunnen handmatig worden verzameld, zoals tijdens inspecties. Soms zullen ze ook met behulp van een tool worden bepaald. Een voorbeeld is een statische analysetool die de codestructuur analyseert. Naast het verzamelen van gegevens wordt soms ook de meting zelf ondersteund door een hulpmiddel. Er is dan sprake van een meettool. Deze berekent de waarden voor de metrieken die worden toegepast. Meestal zijn het tool om de gegevens te bepalen –hiervoor bijvoorbeeld: de statische analysetool- en het meettool geïntegreerd.

In deze subparagraaf is het begrip meten uitgewerkt aan de hand van de begrippen entiteit, de eigenschap van de entiteit, de waarde van de eigenschap en de schaal (of de eenheid) waarin

de waarde wordt uitgedrukt. Ook is aangegeven dat er metrieken nodig zijn om te kunnen meten. Metrieken betreffen zowel het meetvoorschrift als de schaal waarop de waarde wordt uitgedrukt. In de volgende subparagraaf gaan we verder in op metrieken.

2.2.2 Code- en vraaggebaseerde metrieken

In de meetliteratuur wordt een groot aantal metrieken beschreven, zie bijvoorbeeld in (Halstead, 1977), (Kafura en Reddy, 1987), (Chidamber en Kemerer, 1994), (Li en Henry, 1995), (Wilkie en Hylands, 1998). Soms worden metrieken aangeduid als object-geörienteerde, dan weer als codemetrieken, structuur-, ontwerp- of specificatie-metrieken. In de loop van dit proefschrift zullen we constateren dat er ten aanzien van beoordelingsmethoden twee benaderingen zijn te onderkennen, namelijk: codemetriek en vraaggebaseerde beoordelingen. Het onderscheid heeft te maken met de soorten metrieken die in de benaderingen worden toegepast. In deze paragraaf komt het onderscheid tussen de twee soorten metrieken aan de orde.

Een *codemetriek* betreft een metriek die in een waarde resulteert op een interval-schaal. Voorbeelden zijn ‘aantal regels code’ en ‘aantal niveaus van nesting in de code’. Een vraag daarentegen is een metriek die leidt tot een kwalificatie (op nominale of ordinale schaal). Er is veelal sprake van een verzameling vragen die betrekking heeft op één eigenschap. Bepalend voor het onderscheid tussen codemetrieken en vragen is dat laatstgenoemde worden beantwoord door mensen. Hierdoor heeft waarde veelal een kwalitatief karakter. De waarde resulterend uit een codemetriek is vaak kwantitatief van karakter.

Naast de dimensie kwalitatief versus kwantitatief, spelen er nog drie dimensies bij het onderscheid tussen beide metrieksoorten, namelijk:
- direct versus indirect,
- extern versus intern en
- objectief versus subjectief.

Deze drie dimensies worden in deze subparagraaf verder uitgewerkt.

Direct versus indirect

Het onderscheid tussen directe en indirecte meting betreft de manier waarop de waarden worden toegekend aan de eigenschappen. Soms kan er een directe afbeelding van eigenschap naar waarde worden gemaakt. In het geval van een directe metriek vereist de meting geen andere eigenschappen of entiteiten. Codemetrieken, zoals de lengte van de broncode, gemeten in Regels code (Eng.: lines of code) zijn vaak directe metrieken.

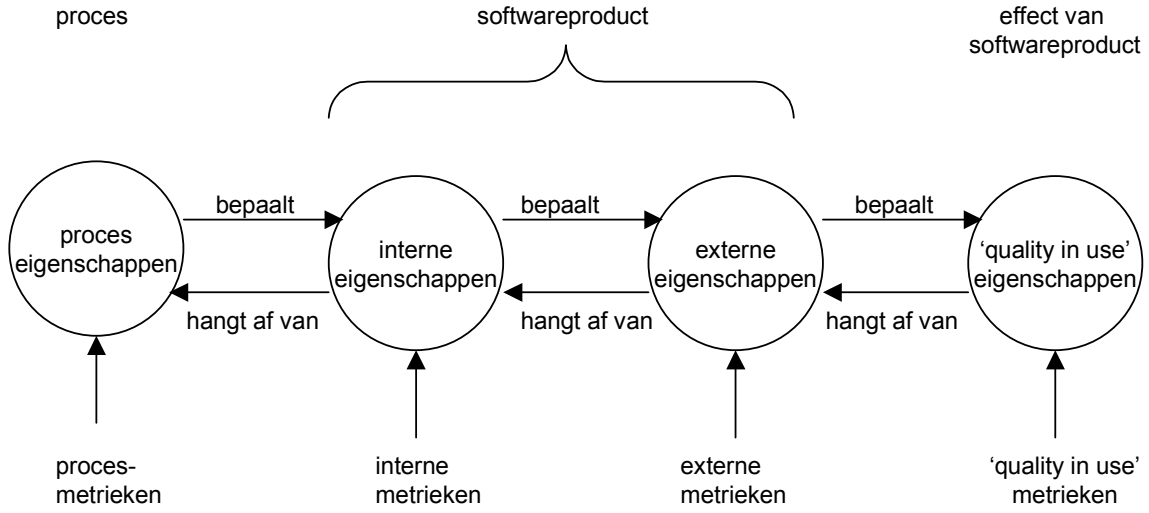
Indirecte metrieken worden toegepast als de eigenschap niet in één keer kan worden gemeten. Er zijn dan andere metrieken nodig waarvan de metingen gecombineerd worden tot

de indirecte metriek. De waarden van de metingen aan de eigenschappen worden dan gecombineerd tot één waarde. Een voorbeeld van een indirecte metriek is de codemetriek Halstead's E (Effort) om inspanning te meten. Het meetvoorschrift luidt: E = V/L. Hierbij zijn V en L de metrieken die eerst bepaald moeten worden voordat de waarde van E is te bepalen. De waarde van E wordt samengesteld uit die van V en L. Een ander voorbeeld van een indirect metriek is de vraaggebaseerde metriek om de leesbaarheid van documentatie te bepalen. Er worden dan vragen gesteld, zoals: 'is er een inhoudsopgave aanwezig?', 'zijn alle termen gedefinieerd?'. De antwoorden op deze vragen leveren tezamen een score voor de leesbaarheid opleveren. Voor een uitgebreide behandeling van consequenties van en problemen met indirecte metrieken wordt verwezen naar (Fenton en Pfleeger, 1996) en (Abran en Robillard, 1996).

Extern versus intern

Het onderscheid tussen externe en interne metrieken heeft betrekking op de soort eigenschappen die ze meten. Interne eigenschappen worden beschreven in termen van het product. Voorbeelden zijn de structuur van de code of het aantal regels code. Externe eigenschappen daarentegen, worden bepaald door het gedrag van het product. Voorbeelden zijn de onderhoudbaarheid of de functionaliteit van het systeem.

Om de eigenschappen te meten zijn metrieken nodig. De ISO 9126-standaard onderkent in de bijlages (delen II en III) dan ook interne en externe metrieken. Het verband tussen de metrieken en de eigenschappen wordt in onderstaande figuur geïllustreerd.



Figuur 2.3 Benaderingen voor software-kwaliteit. Afgeleid uit Bevan (1997)

De figuur wijst behalve op de relatie tussen (externe en interne) eigenschappen en metrieken ook op twee andere zaken, namelijk:

– dat de externe eigenschappen worden bepaald door de interne eigenschappen en

– dat er ook eigenschappen én metrieken voor proces en ‘quality in use’ zijn.

Het eerste punt betreft de visie dat interne kwaliteit resulteert in externe kwaliteit. Een voorbeeld is de bewerking dat ‘een goede structuur van de code resulteert in beter onderhoudbare code’. Deze visie wordt vaak aangehangen. Echter, dergelijke algemeen geldende relaties zijn tot op vandaag de dag niet bewezen. Zo geven Kitchenham en Pfleeger (1996) aan dat ‘...more research is needed to confirm that internal quality assures external quality’. Boegh e.a. (1999) zeggen hetzelfde als zij beweren dat ‘Most software engineers believe that properties of the intermediate products of the development process affect final product behaviour, but there is little scientific evidence of this link and no successful examples of using it to predict or control final product behaviour’.

Het tweede punt betreft de visie dat ook proces- en ‘quality in use’ kwaliteit van belang zijn voor een beoordeling. Voor beide zijn metrieken nodig om de waarde ervan vast te stellen. De rol van proceskwaliteit voor productbeoordelingen is bij de indeling van entiteiten al aan de orde gekomen. ‘Quality-in-use’ is een begrip dat is gelanceerd tijdens de ontwikkeling van de nieuwe versie van de ISO 9126-standaard (Azuma, 1996), (Bevan, 1997). Het wordt gedefinieerd als ‘the extent to which a product used by specified users meets their needs to achieve specified goals with effectiveness, productivity and satisfaction in a specified context of use’ (Bevan, 1997). Quality-in-use betreft het perspectief waarmee de gebruiker naar het product kijkt.

Voor de relatie tussen product en interne eigenschappen en de relatie tussen externe en ‘quality-in-use’-eigenschappen geldt dezelfde opmerking als voor de relatie tussen externe en interne eigenschappen: de relaties zijn grotendeels onbekend. Er is weinig bekend hoe de eigenschappen elkaar beïnvloeden. In de praktijk mag dan vaak het adagium worden gehanteerd dat productkwaliteit wordt bepaald door het proces, dergelijke relaties zijn nog niet wetenschappelijk aangetoond. Zie bijvoorbeeld Schneidewind (1998) of Van Solingen (2000).

Omdat de relaties tussen proces-, interne, externe en ‘quality-in-use’ eigenschappen nog grotendeels onbekend zijn, kan hiervan geen gebruik worden gemaakt. Het gevolg is dat metrieken moeten worden afgestemd op het type eigenschap. Zo zijn ‘quality-in-use’ metrieken nodig om ‘quality-in-use’ eigenschappen te meten. Er kan niet worden teruggevallen op interne metrieken.

Codemetrieken zijn geschikt voor het meten van interne eigenschappen. Vraaggebaseerde metrieken zijn met name geschikt voor het meten van externe en ‘quality-in-use’ eigenschappen. Het gedrag van systemen kan goed worden ingeschat door de mensen, die ervaring met het systeem hebben, hiernaar te vragen. Voor het gebruik van het systeem geldt dezelfde redenering.

Objectief versus subjectief

Het onderscheid tussen objectief en subjectief heeft te maken met het meetvoorschrift: hoe de meting wordt uitgevoerd. Tijdens een objectieve meting wordt de bepaling van de waarde niet beïnvloed door de gevoelens of opinies van een persoon. De meting is persoonsonafhankelijk. De bepaling van de waarde hangt alleen af van de entiteit die gemeten wordt; niet van het gezichtspunt van waaruit de meting plaatsvindt. Een subjectieve inschatting wordt bepaald door de persoon, het gezichtspunt van waaruit de meting wordt uitgevoerd.

Codemetrieken, zoals aantal regels code en cyclomatische complexiteit (McCabe, 1976), worden veelal als objectief gekwalificeerd. Overigens is dat niet zo hard als het lijkt. Zo zijn er complicaties ten aanzien van van de objectiviteit van de metriek 'aantal regels code'. Er zijn namelijke verschillende uitvoeringen van de metriek bekend. Dit komt omdat de vraag 'wat is een regel code?' op verschillende manieren kan worden geïnterpreteerd. Wordt het commentaar tot een regel code gerekend of niet? Vandaar dat er in het verleden een procedure is opgesteld waarin is gestandaardiseerd wat regels code zijn (Park, 1992). Voor een groot aantal andere codemetrieken is van dergelijke standaardisatie nog geen sprake. De metrieken zijn daarmee niet echt objectief, zoals vaak wordt aangenomen. In verhouding met veel vraaggebaseerde metrieken zijn ze echter wel als objectief op te vatten. De subjectieve inschatting is gerelateerd aan de vraaggebaseerde metrieken. Er wordt immers aan personen (beoordelaar, gebruiker, leverancier van product) gevraagd wat men ervan vindt. Door dit subjectieve karakter worden vragen nogal eens als minder gewaardeerd. Zo vereist de ISO 14598-5 (1998)-standaard objectiviteit. De vraaggebaseerde metrieken hebben echter op andere dimensies een meerwaarde.

Codemetriek en vraaggebaseerde metriek

Met de behandeling van deze dimensies is een begrippenkader geformuleerd waarmee het onderscheid tussen codemetriek en vraaggebaseerde metrieken is uitgewerkt. In het bovenstaande is per dimensie al aangegeven hoe deze betrekking hebben op de codemetriek of de vraaggebaseerde metriek. Met onderstaande tabel wordt een overzicht gegeven.

Tabel 2.3 Identificatie van code en vraaggebaseerde-metriek

	Codemetriek	Vraaggebaseerde-metriek
Waarde (op schaal)	Kwantitatief	Kwalitatief
Waarde toekenning	Direct, indirect	Indirect
Eigenschappen	Intern	Quality-in-use, extern
Meetprocedure	Objectief	Subjectief

Deze tabel moet kolomsgewijs worden gelezen. Voor de kolom codemetrieken volgt dan dat er een kwantitatieve waarde resulteert. De waardetoekenning kan zowel direct als indirect plaatsvinden. De metingen richten zich voornamelijk op interne eigenschappen van het

product en de meetprocedure is objectief. Vraaggebaseerde metrieken leveren daarentegen een kwalitatieve waarde op. De waardetoekenning is veelal indirect. De metingen richten zich voornamelijk op Quality-in-use en externe eigenschappen van het product. De meetprocedure is subjectief.

Deze indeling is niet geheel orthogonaal: het is mogelijk dat een codemetriek een kwalitatieve waarde oplevert, ook zijn er vraaggebaseerde metrieken die zich richten op de interne eigenschappen van het product. De indeling is dan ook geen strikt beschrijvingskader. Het heeft alleen tot doel om voor het grote aantal beoordelingsmethoden tot een ordeningskader te komen. In paragraaf 3.4 komt dit verder aan de orde.

2.2.3 Eisen aan de metrieken

In dit proefschrift worden twee eisen aan metrieken gesteld, namelijk dat ze reproduceerbaar en valide moeten zijn. Beide eisen worden hieronder toegelicht.

Reproduceerbaarheid

Een meting is reproduceerbaar als het herhaald uitvoeren van de meting door dezelfde of andere personen dezelfde of vergelijkbare waarden oplevert. Reproduceerbaarheid heeft betrekking op de meetprocedure en daarmee op het onderscheid tussen objectief en subjectief.

De vraaggebaseerde metrieken betreffen subjectieve inschattingen (Eng.: ratings). Om deze toch te kunnen beschouwen als metingen, dienen ze reproduceerbaar te zijn. Hiervoor moeten de verschillen in de beantwoording van de vragen, de variantie, beperkt worden. Een aanpak is het gebruik van ‘gesloten’ vragen. Dit zijn vragen met een beperkte verzameling voorgedefinieerde antwoordmogelijkheden. De beoordelaar kiest één van de antwoordmogelijkheden bij het beantwoorden van een vraag. Een herhaalde beantwoording kan leiden tot een andere keuze, maar het is niet mogelijk om andere dan de gekozen mogelijkheden te kiezen. Het aantal keuzemogelijkheden wordt daarmee beperkt. Richtlijnen om checklisten op een dergelijke manier te construeren zijn te vinden in (Punter, 1997).

Het toepassen van ‘gesloten’ vragen wordt als een verbetering beschouwd wat betreft het bereiken van bepaalde mate van reproduceerbaarheid van de beoordeling. Veel van de door ons bekeken vraaggebaseerde beoordelingsmethoden volgen deze aanpak niet. Maar ook als de methode ‘gesloten’ vragen toepast, zijn de problemen nog niet voorbij. De variantie is nog altijd hoog. Illustratief hiervoor is een experiment dat is uitgevoerd binnen de Deense beoordelingsorganisatie Delta (Andersen en Kyster, 1994). Tijdens een softwareproduct-beoordeling werden ‘gesloten’ vragen afzonderlijk beantwoord door twee ervaren software beoordelaars. De antwoorden zijn achteraf met elkaar vergeleken. In totaal zijn 115 ‘gesloten’ vragen uit 9 checklists beantwoord. Per checklist werden tussen 5 en 27 vragen gesteld. Na analyse van de antwoorden bleek dat 40% van de gestelde vragen verschillend

werd beantwoord. Deze hoge mate van variantie maakt dat ook de reproduceerbaarheid van benaderingen met ‘gesloten’ vragen beperkt is.

Andere maatregelen in het kader van het bereiken van reproduceerbare vraaggebaseerde beoordelingen worden gegeven door Xenos en Christostodoulakis (1997). Zij pleiten ervoor om controlevragen toe te voegen, vragen te herhalen door ze anders te formuleren en vragen te herhalen met verschillende responsiemogelijkheden). Andere auteurs (Tsukumo e.a., 1996), (Fusaro e.a., 1998) komen met vergelijkbare voorstellen.

Ondanks al deze maatregelen zal er altijd variantie in de beantwoording van vraaggebaseerde metrieken bestaan. Het is dan ook zaak dit te onderkennen en zich bewust te zijn van de beperkingen van een dergelijke aanpak. Dit kan door het bepalen van de vereiste reproduceerbaarheid of toegestane variantie per beoordeling. Op deze manier kan men grenzen stellen en het nadeel –beperkte reproduceerbaarheid– combineren met het voordeel –de bredere scope en eenvoudiger opstelling– van vragen.

Validiteit

Het begrip validiteit betreft de vraag ‘wordt de eigenschap gemeten die we willen meten?’ Het gaat om de geldigheid van de metriek.

Een valide meting vereist dat de relatie tussen eigenschap(pen) en metriek(en) is bepaald. In de literatuur wordt hiervoor de term *empirisch model* (Eng.: empirical model) gebruikt (Fenton en Pfleeger, 1996). Dit model beschrijft hoe de metriekwaarden worden toegewezen aan de eigenschappen die worden gemeten. Een voorbeeld is een model dat aangeeft dat de metrieken ‘Aantal Regels code’, ‘Cyclomatische complexiteit’ en ‘Aantal veranderingsvoorstellen’ de eigenschap ‘analyseerbaarheid’ bepalen. Het model zal ook de gewichten van elk van de toegepaste metrieken en de normwaarden definiëren. Dergelijke modellen kunnen vooraf aan het meten worden opgesteld (a priori).Ook is het mogelijk dat ze tijdens het interpreteren van de meetresultaten worden geformuleerd. Wannéér het model wordt opgesteld –a priori of a posteriori– hangt af van de mogelijkheid om een model voor de te meten eigenschap op te stellen.

We constateren dat kennis over de relaties tussen metrieken en te meten eigenschappen beperkt is. Er is en wordt het nodige onderzoek verricht, voorbeelden zijn te vinden in (Abreu e.a., 1994), (Briand e.a., 1999b), (Koshgoftaar en Allan, 1998), (Shepperd en Cartwright, 1999) en (Sahraoui en Azar, 1999)[8]. Dit onderzoek heeft echter nog niet geleid tot algemeen geldige wetten waarin het gedrag en/of de structuur van software en hoe dit te

[8] Deze verhandeling over empirische modellen geeft aan dat de validiteit van metrieken niet kan worden geclaimd door alleen statische relaties tussen de metrieken te bepalen. Dergelijke correlaties impliceren geen oorzaak gevolg relatie, maar leiden mogelijk tot zogenaamde ‘shotgun correlations’ (Courtney en Gustafson, 1993).

meten wordt beschreven. Vaak is er daardoor sprake van onenigheid over de relaties tussen metriek en eigenschap. Het meest sprekende voorbeeld hiervan is de discussie over metrieken voor het bepalen van de omvang van code (Eng.: size of code): zijn Regels code of juist Functiepunten de beste voorspeller? Deze discussie duurt al enige tijd en er lijkt nog geen eind aan te komen (DeMarco, 1995), (Symons, 1998), zolang de relaties gebaseerd zijn op geloof in plaats van algemeen erkende relaties.

Wij gaan er zelfs vanuit dat wetten, waarin dergelijke relaties beschreven worden, nooit algemeen geldend zullen worden. Dit komt omdat de relaties alleen bruikbaar zijn in de context van specifieke softwareproducten. Dit is weer het gevolg van de context afhankelijkheid van software-kwaliteit

Het lijkt ons dan ook effectiever om tot consensus te komen binnen de groep waar de metrieken worden toegepast, dan te streven naar algemeen geldige wetten over de relaties tussen metrieken en eigenschappen De empirische modellen dienen dan door de groep mensen –bijvoorbeeld een team engineers– worden herkend en geaccepteerd te worden. We duiden dit aan als *intersubjectiviteit*: binnen de groep waar de metingen worden uitgevoerd is er overeenstemming welke metriek X de eigenschap Y meet. De subjectieve opinies van de mensen uit de groep wordt gecombineerd tot een binnen de groep geldige opinie.

Het streven naar intersubjectiviteit komt in het proefschrift een aantal keren aan de orde. Zo wordt in de casestudie in hoofdstuk 4 gerapporteerd over een experiment waarin de engineers van het betreffende product de kwaliteit van de software bepalen. Ook in het ontwerp (hoofdstuk 6) komt het aan de orde.

2.3 Samenvatting

In deze paragraaf is uitgewerkt wat we onder het meten aan software verstaan. Meten is het toekennen van waarden aan eigenschappen van een softwareproduct. De waarde wordt uitgedrukt als een eenheid op een schaal. Om de meting uit te voeren zijn metrieken nodig, deze betreffen zowel de schaal als het meetvoorschrift van de meting.

Vervolgens is ingegaan op het onderscheid tussen twee soorten metrieken, namelijk code- en vraaggebaseerde metrieken. Het onderscheid is uitgewerkt aan de hand van vier dimensies, namelijk: kwalitatief versus kwantitatief, direct versus indirect, extern versus intern en objectief versus subjectief. Tot slot zijn er twee eisen voor metrieken opgesteld, namelijk reproduceerbaarheid en validiteit. Hierbij zijn de mogelijkheden en moeilijkheden om aan deze eisen te voldoen kort besproken.

3. Analyse van beoordelingsmethoden

In dit hoofdstuk wordt een analysekader opgesteld om bestaande beoordelingsmethoden te analyseren. Met dit kader wordt onder andere bepaald in hoeverre beoordelingsmethoden aandacht geven aan de besturing van een beoordelingsproces. De resultaten van de analyse, die met behulp van dit kader is uitgevoerd, worden in de paragrafen 3.5 en 3.6 gepresenteerd. Vervolgens wordt een aangescherpte probleemdiagnose van het onderzoek opgesteld. Dit is een verdere uitwerking van de probleemstelling zoals gegeven in hoofdstuk 1.

3.1 Inleiding

In hoofdstuk 1 is aangegeven dat beoordelingen van softwareproducten nogal eens leiden tot ontevredenheid over de bereikte resultaten. In dat hoofdstuk is ook aangegeven dat er in dit onderzoek vanuit wordt gegaan dat deze ontevredenheid met name voortkomt uit het beoordelingsproces. Als dit proces niet goed is opgezet en niet goed wordt uitgevoerd, dan kan en mag men geen succesvolle resultaten verwachten.

In dit onderzoek kijken we vanuit een bedrijfskundige optiek naar beoordelingsprocessen. Daarbij staat de besturing van het proces centraal. Bij het uitwerken van het besturingsperspectief gebruiken we concepten uit de systeemtheorie (De Leeuw, 1980), (Kramer en De Smit, 1987). De systeemtheorie geeft aan het dat een proces opgevat kan worden als een systeem met invoer en uitvoer (De Leeuw, 1980). De invoer van een beoordelingsproces bestaat onder andere uit de verwachtingen van de klant over de beoordelingen en uit informatie over een product. De uitvoer van het beoordelingsproces betreft het resultaat van de beoordeling. Dit is het oordeel of het product wel of niet voldoet. Het proces moet zodanig zijn dat invoer resulteert in een uitvoer die een uitspraak doet over het beoordeelde. Hiervoor is besturing nodig. Een proces wordt immers niet zomaar uitgevoerd, maar is gericht op het bereiken van een doel. Het vaststellen van het doel van een beoordeling is dan ook één van de aspecten van besturing.

In dit hoofdstuk wordt nagegaan in hoeverre beoordelingsprocessen bestuurd worden. Dit wordt gedaan door beoordelingsmethoden uit de literatuur te analyseren. Onder een beoordelingsmethode verstaan we een stappenplan voor het uitvoeren van een softwareproduct-beoordeling inclusief de daarbij gehanteerde metrieken. In hoofdstuk 1 zijn reeds een aantal methoden genoemd (fail-safe inspectie, product audit, de SUMI-methode voor usablity testing en conformity assessment).

Voor het analyseren van de beoordelingsmethoden wordt eerst een analysekader opgesteld (paragraaf 3.2). Vervolgens wordt dit kader toegepast om diverse methoden te analyseren. De

resultaten daarvan worden gepresenteerd in de paragrafen 3.3 tot en met 3.5. Tenslotte wordt de probleemstelling aangescherpt op basis van deze analyseresultaten (paragraaf 3.6).

3.2 Analysekader

3.2.1 Voorwaarden voor effectieve besturing

Het analysekader is gebaseerd op de voorwaarden van effectieve besturing zoals de systeemtheorie die formuleert voor een besturingssituatie. Een besturingssituatie wordt in de systeemtheorie beschreven aan de hand van drie elementen. (De Leeuw, 1974, 1980). Het betreft:

- Bestuurd systeem (BS) – het (deel)systeem dat bestuurd wordt. Het betreft het proces waarbinnen input wordt omgezet in beoogde output. Dit proces is het geheel van activiteiten en de relaties daartussen.
- Besturend orgaan (BO) – het deelsysteem dat het bestuurd systeem bestuurt.
- Omgeving (O) – dit is de omgeving van zowel het BO als het BS. Deze omgeving beïnvloedt het gedrag van BS en BO, onder andere door de doelstelling van het beoordelingssysteem te beïnvloeden. Klant en leverancier van een softwareproduct behoren bijvoorbeeld tot de omgeving.

De Leeuw (1974, 1980) geeft voorwaarden waaraan het besturend orgaan moet voldoen om effectief te kunnen sturen. Deze worden aangeduid als *voorwaarden voor effectieve besturing*. Het betreft:

1. Doel – het besturend orgaan moet ten aanzien van het bestuurd systeem een doel specificeren, dat richtsnoer is bij de besturing.
2. Model – het besturend orgaan moet over een adequaat model van het bestuurd systeem beschikken.
3. Informatie – het besturend orgaan moet beschikken over voldoende informatie over de toestand van het bestuurd systeem en daarop inwerkende omgevingsgrootheden,
4. Voldoende besturingsvariëteit – het besturend orgaan moet beschikken over voldoende stuurmaatregelen om het systeem te sturen.

Elk van deze voorwaarden komt hieronder aan de orde.

Ad 1: dat er een doel vastgesteld moet worden, volgt uit het uitgangspunt dat besturing een doelgerichte beïnvloeding beoogt. De systeemtheorie gaat ervan uit dat er altijd een doel is. De Leeuw merkt op dat doelen niet noodzakelijk expliciet, constant en compleet hoeven te zijn om te kunnen spreken van besturing. Het gaat er evenwel om dat het resultaat wordt getoetst aan een doel. De Leeuw zegt daarover: ‘besturing noopt tot het minimaal aanwezig zijn van een evaluatiemechanisme ter beoordeling van de effecten van de beïnvloeding.

Anders is van gerichte beïnvloeding geen sprake' (De Leeuw, 1980). Eigenlijk staat hier dat terugkoppeling noodzakelijk is, anders kan men niet evalueren. Terugkoppeling betreft zowel de resultaten van deelprocessen als van het beoordelingsproces als geheel.

Ad 2: er is een model nodig omdat besturing impliceert dat het besturend orgaan ingrijpt op het bestuurd systeem door middel van stuurmaatregelen. Het besturend orgaan moet dus inzicht hebben hoe een product in elkaar zit en welke ingrepen waar mogelijk en wenselijk zijn. Dit inzicht wordt geformuleerd middels een model. Als dit inzicht ontbreekt dan is gerichte sturing onmogelijk. De Leeuw geeft aan dat het niet vereist is om een compleet model bij aanvang van de besturing te hebben. Zo'n model kan gaandeweg worden opgebouwd en verfijnd.

Ad 3: dat een besturend orgaan moet kunnen beschikken over voldoende informatie, volgt uit de vorige voorwaarde. Het model kan immers alleen goed worden gehanteerd als er informatie over de actuele toestand van het systeem en de relevante omgevingsfactoren aanwezig is (Kramer en de Smit, 1987). Het model bepaalt daarmee wat relevante informatie is over het bestuurd systeem en de omgeving. Informatie is in die zin een afgeleide van het model.

Ad 4: dat een besturend orgaan voldoende besturingsmaatregelen ter beschikking moet hebben, lijkt triviaal (Kramer en De Smit, 1987). Immers als een besturend orgaan geen adequate maatregelen kan treffen, zal de besturing niet effectief zijn. De voorwaarde is echter wel degelijk relevant, omdat deze voorwaarde impliceert dat het aantal stuurmaatregelen in een redelijke verhouding moet staan tot de variëteit aan omstandigheden die zich kunnen voordoen.

Naast deze vier voorwaarden wordt soms verwerkingscapaciteit als een voorwaarde genoemd; bijvoorbeeld in (De Leeuw, 1980). Het betreft de verwerkingscapaciteit om informatie, over omgevingssysteem en doelstelling, met behulp van het model om te zetten in een effectieve stuurmaatregel. Deze voorwaarde is met name relevant in situaties waar snelheid van informatieverwerking in milliseconden moet gebeuren bijvoorbeeld bij chemische procesregelingen die bij traag reageren kunnen leiden tot catastrofes. Omdat beoordelen niet tijdkritisch is achten wij deze voorwaarde niet van belang voor de besturing van beoordelingsprocessen. Daarom wordt de voorwaarde verder weg gelaten.

3.2.2 Van voorwaarden naar analysekader

De voorwaarden voor effectieve besturing vormen de basis van ons analysekader. We gebruiken hierbij met name twee van vier voorwaarden, namelijk: 'model' en 'besturingsvariëteit'. De voorwaarden 'doel' en 'informatie' komen niet expliciet in het kader aan de orde maar worden geverifieerd bij het beoordelen van de diverse methoden.

Uitwerking van voorwaarde 'model'
De beschikking over een model is als voorwaarde gesteld omdat zo'n model inzicht geeft in de mogelijkheden om te sturen. Voor een beoordelingsmethode betekent dit, dat er op zijn minst moet zijn aangegeven:
- uit welke activiteiten het beoordelingsproces bestaat en
- hoe de verschillende activiteiten samenhangen.

Bij de *beschrijving van activiteiten* gaat het erom dat er is aangegeven welke activiteiten er zijn en wat ze beogen. Zo kan een methode aandacht besteden aan het specificeren van eigenschappen voor een softwareproduct en aan het vaststellen van normwaarden. De beschrijving van activiteiten moet zodanig zijn dat aangegeven wordt hoe de invoer van een activiteit tot uitvoer leidt.

De *samenhang van de activiteiten* ofwel *processtructuur* geeft aan hoe de activiteiten van een proces met elkaar samenhangen. Ofwel: welke invoerstromen een activiteit nodig heeft van andere activiteiten en welke andere activiteiten de uitvoer van een activiteit gebruiken. Het heeft onder meer te maken met de volgorde van activiteiten en de interacties tussen deze activiteiten.

Uitwerking van voorwaarde 'besturingsvariëteit'
Besturingsvariëteit houdt in dat een besturend orgaan maatregelen moet kunnen treffen om bij te sturen. Ook het aantal maatregelen moet in een redelijke verhouding staan tot de variëteit aan omstandigheden. Een en ander houdt in dat de bestuurbaarheid van een proces toeneemt naarmate er meer mogelijkheden zijn om een proces bij te sturen. Voor de analyse van de beoordelingsmethoden onderkennen we drie soorten mogelijkheden om te sturen, namelijk:
- aansturen van de soort, volgorde en de interactie van activiteiten,
- afwegen van doel en middelen en
- terugkoppelen en eventueel bijstellen.

Aansturen van het proces houdt in dat er per beoordeling eerst wordt bekeken welke activiteiten uitgevoerd moeten worden. Er hoeft namelijk niet altijd een standaardverzameling activiteiten te worden uitgevoerd. Dit is immers afhankelijk van de situatie; zie het uitgangspunt bij dit onderzoek in hoofdstuk 2 ten aanzien van de contingentiebenadering. Het is belangrijk om te bepalen welke activiteiten noodzakelijk zijn en welke kunnen worden overgeslagen. Voor een beoordelingsproces betekent dit bijvoorbeeld dat de activiteit 'definiëren kwaliteitsmodel' niet wordt uitgevoerd, als al bekend is wat de te beoordelen eigenschappen zijn. Naast het bepalen welke activiteiten worden uitgevoerd, is het belangrijk om te overwegen in welke volgorde dit moet gebeuren en tot op welk detailniveau de activiteiten moeten worden uitgevoerd. Het aspect aansturen van een proces duiden we ook aan als *situatie-afhankelijk inrichten van een proces* of met de

Engelse term: '*customising'*: het afstemmen van het proces op het beoordelingsdoel en op de beoordelingscontext.

Het afwegen van doel en middelen betreft een tweede categorie stuurmaatregelen. Het gaat hierbij om de vraag of doel en de inzet van middelen in een redelijke verhouding tot elkaar staan. Als dat niet zo is dan zal men doel, middelen of beide moeten aanpassen. Het afwegen van doel en middelen wordt ook aangeduid als het *balanceren van doel en middelen*.

De derde categorie van stuurmaatregelen heeft te maken met het organiseren van *terugkoppeling (feed back) en het bijstellen van een proces*. Activiteiten moeten worden bijgesteld als er tijdens de uitvoering van een proces blijkt dat de beoogde output niet gehaald wordt. Vervolgens moet opnieuw worden bepaald welke activiteiten dan uit te voeren en er moet opnieuw een afweging tussen doel en middelen plaatsvinden.

Het voorgaande is vertaald naar vijf aspecten binnen ons analysekader. Elk van die aspecten geeft aan waaraan een beoordelingsmethode zou moeten voldoen, wil er sprake zijn van een effectieve besturing van een beoordelingsproces. De vijf aspecten zijn respectievelijk:
- overzicht van activiteiten,
- processtructuur,
- aansturing van beoordelingsproces,
- afwegen van doel en middelen,
- terugkoppelen en bijstellen van proces

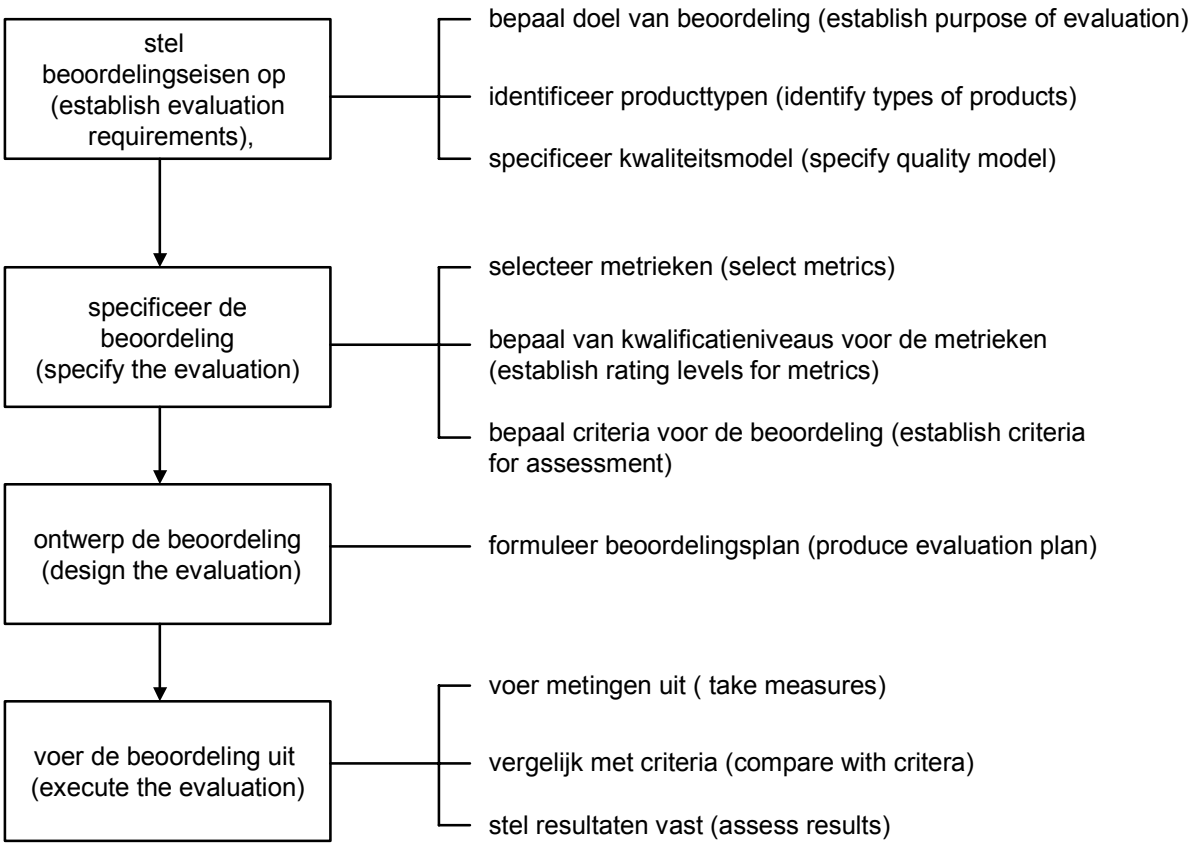
3.2.3 Het aspect: overzicht van activiteiten

Een beoordelingsmethode moet een overzicht geven van activiteiten en per activiteit aangeven hoe van invoer naar de beoogde uitvoer van elke activiteit te komen. Probleem is nu dat elke methode zijn eigen terminologie en manier kent om activiteiten te beschrijven. Om de activiteiten van elk van deze methoden te kunnen vergelijken, wordt daarom uitgegaan van de beschrijving die ISO 14598 geeft van een beoordelingsproces.

ISO 14598 activiteiten

ISO 14598-1 geeft een algemeen overzicht welke activiteiten standaard tot een beoordelingsproces behoren. De ISO-standaard onderkent vier deelprocessen, namelijk:
- vaststellen van beoordelingseisen (establish evaluation requirements),
- specificeren van de beoordeling (specification of the evaluation),
- ontwerpen van de beoordeling (design of evaluation),
- uitvoeren van de beoordeling (execute the evaluation).

De standaard geeft aan dat de vier deelprocessen achter elkaar moeten worden uitgevoerd. Elk deelproces bestaat uit een aantal activiteiten. De volgende figuur vermeldt de deelprocessen en activiteiten.



Figuur 3.1 Beoordelingsproces volgens ISO 14598-1 (1999)

Aan de linkerzijde van de figuur staan de vier deelprocessen. De pijlen tussen de deelprocessen geven aan dat ze op elkaar volgen. Het deelproces ‘specificeren van de beoordeling’ volgt dus op ‘vaststellen van beoordelingseisen’. Aan de rechter zijde van de figuur zijn de activiteiten van de diverse deelprocessen vermeld.

Hieronder worden de activiteiten kort uitgewerkt. Dit wordt gedaan door voor elke activiteit te beschrijven wat ISO 14598-1 hieronder verstaat.

- *bepaal doel van beoordeling* (establish purpose of evaluation) – er wordt een doel bepaald van de beoordeling. ISO 14598-1 geeft bij de beschrijving van deze activiteit een aantal voorbeelden van doelen, zoals: ‘beoordelen of en wanneer een product op de markt kan komen (gereleased wordt)’ en ' beslissen wanneer een product vervangen moet worden’,
- *identificeer producttypen* (identify types of products) – het product dat beoordeeld moet worden moet worden vastgesteld. ISO 14598-1 geeft aan dat dit afhangt van de levenscyclus waarin het product verkeert en het doel van de beoordeling,

- *specificeer kwaliteitsmodel* (specify quality model) – er wordt bepaald wat de voor het product relevante ISO 9126 kwaliteitskarakteristieken zijn,
- *selecteer metrieken* (select metrics) – het bepalen van de metrieken waarmee de waarden van de kwaliteitskarakteristieken gemeten kunnen worden,
- *bepaal kwalificatieniveaus voor de metrieken* (establish rating levels for metrics) – het per metriek indelen van de meetschaal in verschillende niveaus van tevredenheid. De standaard geeft hierbij aan dat het mogelijk is de schaal op te delen in twee (goed, slecht) of meer categorieën (bijvoorbeeld: niet acceptabel, net acceptabel, acceptabel, meer dan acceptabel). Synoniemen voor kwalificatieniveaus zijn: target values (Kitchenham e.a., 1995a), threshold (Erni en Lewerentz, 1996) en boundaries (Fenton en Pfleeger, 1996),
- *bepaal criteria voor de beoordeling* (establish criteria for assessment) – het opstellen van een procedure om meetresultaten te vertalen naar een score per kwaliteits(sub)karakteristiek. Per (sub)karakteristiek kunnen een aantal metrieken worden toegepast. De procedure geeft dan aan om welke metrieken het gaat en hoe zwaar elke metriek meeweegt in het oordeel over de (sub)karakteristiek. Er worden zo gewichten aan de metrieken toegekend,
- *formuleer beoordelingsplan* (produce evaluation plan) – dit beoordelingsplan beschrijft de uit te voeren acties[9] en de volgorde om ze uit te voeren,
- *voer metingen uit* (take measures) – het toepassen van de metrieken op het softwareproduct. Dit leidt tot meetresultaten,
- *vergelijk met criteria* (compare with critera) – het vergelijken van de meetwaarden, uit voorgaande activiteit met de opgestelde criteria, opgesteld in de activiteit ‘bepalen criteria voor de beoordeling’.
- *stel resultaten vast* (assess results) – het opstellen van een oordeel over het product. Hierbij moet ook worden gelet op de tijd, kosten en inspanning die het uitvoeren van de beoordeling heeft gekost.

De hiervoor gegeven beschrijving van ISO 14598 activiteiten gebruiken wij als kader om de activiteiten van de te bestuderen beoordelingsmethoden te analyseren. De ISO 14598 beschrijving beschouwen we echter niet als strikt maatgevend. Het is dus niet zo dat we een beoordelingsmethode slechter vinden als de beschrijving van activiteiten niet precies conform ISO 14598 is. De reden dat we de ISO-14598 als referentie volgen is dat deze standaard een breed geaccepteerde interpretatie geeft van wat er onder een beoordelingsproces kan worden verstaan.

Wel hebben we soms kritiek op de beschrijving van een aantal activiteiten die de standaard geeft. Zo is in een aantal gevallen geen sprake van een echte activiteit. Er worden

[9] ISO 14598-1 geeft het begrip ‘evaluation methods’. Dit wordt in ISO 14598-5 (1998) gedefiniëerd als ‘procedures describing the action to be performed by the evaluator in order to obtain the result for the specified measurement or verification applied on the specified product components or on the product as a whole’. Wij vatten dit op als zijnde de activiteiten, uit te voeren tijdens deelproces ‘uitvoeren van beoordeling’.

interessante do's en don'ts gegeven, maar het ontbreekt aan een procedure die aangeeft hoe een activiteit uitgevoerd moet worden.

3.2.4 Het aspect: processtructuur

Het voorgaande ging in op de beschrijving van de afzonderlijke activiteiten. De *processtructuur* daarentegen richt zich op de samenhang tussen activiteiten. Het beschrijft alle relaties tussen die activiteiten. Een voorbeeld is te vinden in figuur 3.1. In dit figuur wordt ook de volgorde van deelprocessen aangegeven.

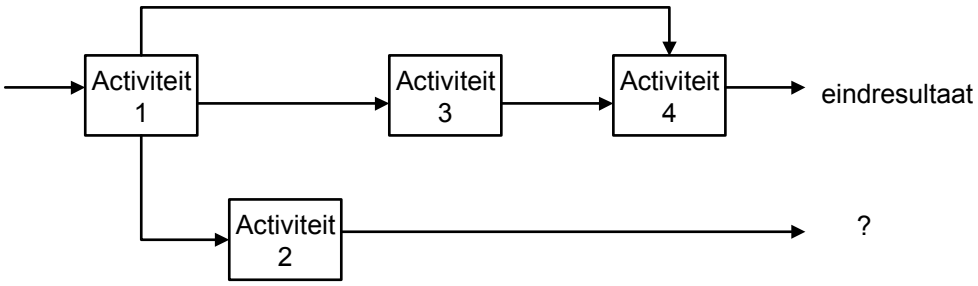
Bij het analyseren van de methoden werken we het aspect processtructuur uit door ons af te vragen of:

- de relaties tussen de activiteiten zijn beschreven en of
- de activiteiten op elkaar aansluiten qua input en output.

Voor elke activiteit is er minimaal één invoerstroom en één uitvoerstroom. Deze zijn gerelateerd aan één of meer andere activiteiten óf aan de omgeving van het proces. Op deze manier zijn activiteiten geschakeld en is er sprake van een keten. Het idee om een proces te zien als een keten is afkomstig uit de logistiek. Het principe wordt bijvoorbeeld door Brombacher gehanteerd om ontwikkelprocessen te analyseren en om te bepalen of alle activiteiten in een (concurrent engineering) proces van belang zijn en goed op elkaar zijn afgestemd (Brombacher, 2000).

Voor de besturing van een proces is het ketenprincipe belangrijk. In het geval van een keten worden de activiteiten niet voor niks uitgevoerd: uitvoer van activiteiten wordt immers toegepast, bijvoorbeeld als invoer voor volgende activiteiten. Een ‘breuk’ in de keten zien we dan ook als een symptoom van slechte besturing van het proces. Er worden dan immers activiteiten uitgevoerd waarvan de resultaten tot ‘niets’ leiden.

Figuur 3.2 geeft daarvan een voorbeeld. De figuur laat zien dat er een activiteit (nummer 2) is gedefinieerd die geen relatie heeft met de andere activiteiten. Dit doet vermoeden dat activiteit 2 geen betekenis voor het proces heeft. Om te verifiëren of dit werkelijk zo is, zal een nadere analyse nodig zijn. Maar het feit dat de activiteit losstaat van de rest van het proces, is al een waarschuwing.



Figuur 3.2 Voorbeeld van een situatie met onvoldoende samenhang tussen activiteiten.

Bij onze analyse van beoordelingsmethoden zullen we steeds vaststellen of er sprake is van een processtructuur bij een beoordelingsmethode. Dit doen we door na te gaan of alle activiteiten van een proces zijn verbonden met één of meer andere activiteiten of het eindresultaat. Ook wordt nagegaan of deze relaties zijn beschreven.

3.2.5 Het aspect: aansturing van het proces

Aansturing van het proces heeft betrekking op drie zaken, namelijk:
- procesinrichting,
- toewijzen van middelen en
- voortgangsbewaking.

Procesinrichting betekent dat er vooraf of tijdens het uitvoeren van het proces wordt nagegaan welke activiteiten in welke volgorde uitgevoerd moeten worden om het doel van een beoordeling te bereiken. Er wordt bijvoorbeeld bepaald of sommige activiteiten misschien parallel moeten worden uitgevoerd. Ook betreft het de overweging bij het ontwerp van het proces of er tijdens de uitvoering moet worden teruggekoppeld.

Het *toewijzen van middelen* (of: middelenallocatie) is van belang om het gedefinieerde proces uit te kunnen voeren. Voor uitvoering zijn immers middelen nodig, deze moeten worden toegewezen. Het gaat hierbij niet om het afwegen van doel en middelen zelf, zie volgende aspect, maar het inplannen en toewijzen ervan. Zo komt bijvoorbeeld wel aan de orde met welke mensen en technieken de activiteiten worden uitgevoerd, maar niet hoe we aan deze middelen komen: op basis van welke kwantiteit en kwaliteit ze zijn gekozen.

Voortgangsbewaking betreft de analyse of de uitvoering van de activiteiten volgens plan verloopt. Hierbij vraagt men zich af of de geplande activiteiten ook feitelijk worden uitgevoerd. Het gaat hier nog niet om de vraag of het beoogde resultaat wordt behaald, dit betreft de terugkoppeling van het proces, zie verderop.

Bij het analyseren van de methoden werken we het aspect aansturing uit door ons steeds af te vragen of er sprake is van:

- procesinrichting – geeft de methode aan welke activiteiten worden uitgevoerd, wordt er aangegeven in welke volgorde dit wordt gedaan en wordt er onderkend dat de activiteiten in een andere volgorde kunnen worden uitgevoerd dan hetgeen is gepresenteerd in de beschrijving van activiteiten en in de processtructuur. De methode kan hier bijvoorbeeld aandacht aan besteden door expliciet de factoren te noemen die bepalend zijn voor het al dan niet uitvoeren van activiteiten.
- toewijzing van middelen – wordt er onderkend dat er middelen aan de diverse activiteiten moeten worden toegewezen?
- voortgangsbewaking – geeft de methode aan dat de voortgang van het proces bewaakt moet worden?

3.2.6 Het aspect: afwegen van doel en middelen

Beoordelen is niet een absoluut proces waarin men kost wat kost een bepaald doel wil bereiken. Niemand zal een 'blanco cheque' willen tekenen om tot een bepaald oordeel over een softwareproduct te komen. Anders geformuleerd niet alleen het doel van een beoordeling is relevant, maar ook de beschikbare tijd en inzet van mensen en middelen om tot een oordeel te komen zijn relevant. Dat vereist dus een (voortdurend) afwegen.

Om te kunnen afwegen is inzicht vereist in zowel doel als middelen. Middelen definiëren we als tijd, mensen en/of technieken (tools) die beschikbaar zijn om activiteiten uit te voeren. Betrokken mensen kunnen zijn beoordelaars en /of gebruikers van het product. Voorbeelden van middelen zijn checklists om bijvoorbeeld Maintainability te bepalen of checklists om in het algemeen kwaliteitskarakteristieken te bepalen.

Bij het afwegen van doel en middelen zijn de begrippen kwaliteit en kwantiteit van belang.

- Kwaliteit geeft aan of de middelen (mensen en technieken) van voldoende kwaliteit zijn. Hebben de betrokken mensen genoeg vaardigheden. Zijn ze bijvoorbeeld ervaren genoeg? Zijn de technieken beproefd? Alternatieve termen voor kwaliteit zijn vaardigheid of competentie (capability),
- Kwantiteit zegt iets over de mate waarin er voldoende tijd, geld, menskracht is. Een alternatieve term voor kwantiteit is capaciteit (capacity)

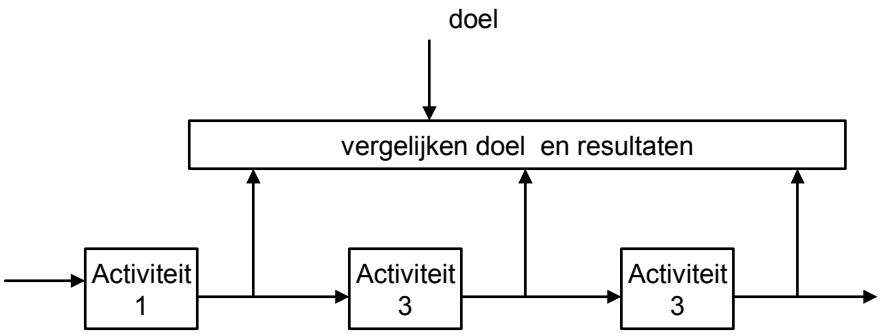
Kwaliteit en capaciteit zijn aspecten die tijdens het afwegen van doel en middelen meespelen. Tijdens een afweging kan bijvoorbeeld blijken dat er wel middelen zijn die voldoen aan het aspect kwaliteit, maar dat er onvoldoende capaciteit beschikbaar is. Tijdens het afwegen dient dan naar alternatieve middelen worden gezocht. Dit gebeurt net zolang totdat er een redelijke verhouding bestaat tussen doel en middelen waarbij ten aanzien van middelen zowel voldaan wordt aan de eisen van capaciteit als aan de kwaliteit.
We duiden dit conform Bemelmans aan als 'de kunst van het balanceren' (Bemelmans, 1998).

Tijdens onze analyse van beoordelingsmethoden zullen we vaststellen of een beoordelingsmethode aandacht besteedt aan het formuleren van een doel en het afwegen daarvan met middelen. Daarbij worden de aspecten kwaliteit en capaciteit gebruikt, om duidelijk te maken waaraan precies aandacht wordt besteed bij een afweging.

3.2.7 Het aspect: terugkoppelen en bijstellen van het proces

Een proces zal moeten worden bijgesteld als er tijdens de uitvoering van activiteiten blijkt dat deze niet resulteren in de verwachte resultaten. Er wordt dan geconstateerd dat het doel niet bereikt zal worden als het proces op de oorspronkelijke manier wordt voortgezet. Dit kan worden veroorzaakt door een verkeerde inschatting wat betreft aansturing van het proces of een verkeerde afweging van doel en middelen.

Om een proces bij te kunnen stellen is terugkoppeling nodig. Eerst moet er immers worden geconstateerd ‘dat er iets misgaat’. Tijdens het terugkoppelen (feedback) wordt de uitvoer van activiteiten vergeleken met het gestelde doel en met de noodzakelijke follow-up van activiteiten. Is de uitvoer van een bepaalde activiteit voldoende als vereiste invoer voor een volgende activiteit? Onderstaande figuur geeft dit terugkoppelingsprincipe voor een proces dat bestaat uit 3 activiteiten. Elke activiteit heeft een in- en een uitvoer. De uitvoer is het resultaat van de activiteit. Deze wordt vergeleken met het gestelde doel.



Figuur 3.3 Drie activiteiten en terugkoppelingsprincipe.

Na terugkoppeling kan blijken dat een proces moet worden bijgesteld. Dit houdt in dat er opnieuw wordt bepaald welke activiteiten worden uitgevoerd en dat er opnieuw een afweging van doel en middelen nodig is. Ofwel: het heroverwegen van de aspecten ‘aansturing van proces’ en ‘afweging van doel en middelen’ is hier aan de orde. Bijsturen kan ook betekenen dat het ambitieniveau van de beoordeling gewijzigd wordt omdat het opnieuw aansturen en afwegen geen zin heeft. Het begrip ambitieniveau wordt hieronder uitgewerkt.

Onder het *ambitieniveau* van een beoordeling verstaan we de mate waarin men een bepaald beoordelingsdoel wil bereiken. Een hoger ambitieniveau zal vaak meer tijd, geld en inspanning kosten. Ambitieniveau kan worden uitgedrukt in termen van meten. Het gaat dan om de mate van reproduceerbaarheid en om de validiteit van een beoordeling. Dit raakt het in hoofdstuk 2 gemaakte onderscheid tussen objectief en subjectief meten. Objectief meten is in principe herhaalbaar in de zin dat het tot dezelfde uitkomsten leidt. Daarmee neemt de validiteit van de metrieken in principe toe. Dat geldt niet voor subjectieve metingen. Herhaling is hierbij bijna onmogelijk en leidt veelal tot andere resultaten. Eén en ander leidt ertoe dat objectieve metingen vaak als van een hoger ambitieniveau worden gekwalificeerd dan subjectieve metingen.

In het geval van een concreet beoordelingsproces kan men starten met de ambitie om alles objectief te meten. Tijdens de uitvoering van het proces kan echter blijken dat dit onhaalbaar is omdat de meting niet uitgevoerd kan worden, of dat er zoveel tijd en inspanning mee is gemoeid dat doel en middelen niet meer in een redelijke verhouding staan. Er rest dan het bijstellen van het ambitieniveau.

Het bijstellen van het ambitieniveau is eveneens gerelateerd aan het mogelijk *veranderen van de omgeving* van een beoordeling. Deze omgeving bestaat onder andere uit de bij een beoordeling betrokken personen, zoals gebruikers en de leverancier van een product. Het doel van een beoordeling wordt eigenlijk door deze omgeving bepaald. Het doel van een beoordeling is immers geen immanente eigenschap van het proces zelf (De Leeuw, 1980). Daardoor kan ook het ambitieniveau dat aan het doel is gesteld, worden veranderd door de omgeving. Anderzijds kan een besturend orgaan ook die omgeving weer beïnvloeden. Zo kan er op de betrokken mensen worden 'ingepraat', dit om hun eisen en verwachtingspatroon ten aanzien van de beoordeling bij te stellen. Er wordt dan bijvoorbeeld aangegeven dat een metriek niet kan worden toegepast en dat er dus andere metrieken gekozen moeten worden, dat de juiste experts niet gevonden kunnen worden, of dat de beoordeling erg duur dreigt te worden, omdat de aan het product gestelde eisen te zwaar zijn. Met dit ‘inpraten op de omgeving’ kan zowel het doel als het ambitieniveau worden bijgesteld.

Tijdens de uit te voeren analyse stellen we vast of er bij beoordelingsmethoden sprake is van bijstellen van een proces door na te gaan of er aandacht wordt besteed aan terugkoppeling of er aandacht wordt besteedt aan het definiëren van een ambitieniveau en of er wordt onderkend dat beide kunnen worden bijgesteld op grond van de terugkoppelingsinformatie.

3.2.8 Samenvatting

Er zijn vijf aspecten geformuleerd die tezamen het kader voor de analyse van de diverse beoordelingsmethoden vormen. De vijf onderkende aspecten zijn:

– Beschrijving van activiteiten,

- Processtructuur,
- Aansturen van het proces,
- Afweging van doel en middelen en
- Terugkoppelen en bijstellen van het proces

Indien een methode voldoet aan elk van de vijf aspecten is er niet automatisch sprake van besturing. Het analysekader is immers met name gebaseerd op twee van de vier voorwaarden voor effectieve besturing: ‘doel’ en ‘informatie’ zijn niet als zodanig geoperationaliseerd. Wel komt de voorwaarde ‘doel’ terug in het aspect ‘afweging van doel en middelen’ en bij het aspect ‘bijstellen van proces’ als ambitieniveau. De voorwaarde ‘informatie’ is impliciet opgenomen in de terugkoppeling bij het aspect ‘bijsturen van proces’.

3.3 Twee beoordelingsbenaderingen

Alvorens het analysekader te gebruiken voor een analyse van diverse beoordelingsmethoden, gaat deze paragraaf in op een ordening van deze methoden. Onderscheiden worden codemetriek- versus vraaggebaseerde beoordelingen. Elke beoordelingsmethode kan tot één van deze twee categorieën worden gerekend of is een combinatie daarvan.

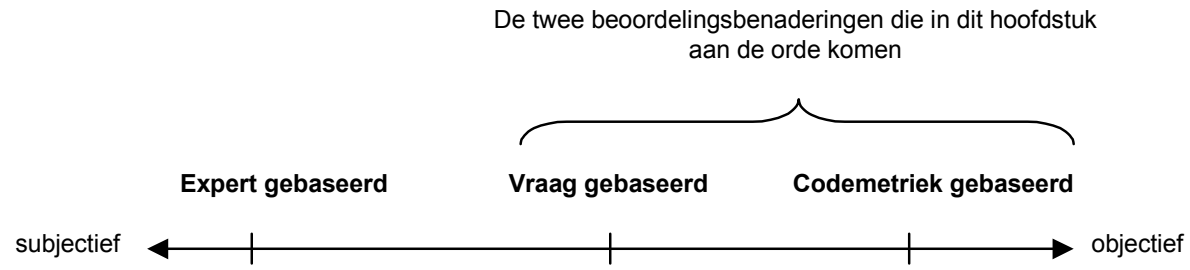
Codemetriek gebaseerde benaderingen betreffen beoordelingsmethoden waarbij codemetrieken centraal staan. Metriekwaarden worden in het algemeen automatisch berekend door een ondersteunend tool. Bij vraaggebaseerde methoden gaat het om beoordelingen die worden uitgevoerd met behulp van vragen die vervolgens beantwoord worden. Deze vragen zijn veelal opgenomen in een checklist.

De indeling vraag- versus codemetriek gebaseerd grijpt terug op het onderscheid tussen de objectieve en de subjectieve manier om waarden vast te stellen.

In het geval van de codemetriek gebaseerde methoden gebeurt dit op een objectieve wijze. Objectief in de zin dat de bepaling van de meetwaarden niet afhangt van de persoon die de metrieken toepast; zie paragraaf 2.2.2. De codemetriek gebaseerde methoden hanteren een strikt persoonsonafhankelijke meetprocedure. Een dergelijke objectieve waardevaststelling levert beter reproduceerbare resultaten op dan de subjectieve wijze.

Naast een objectieve en een subjectieve meetbenadering kunnen we nog een derde benadering onderscheiden: de *expertbenadering*. Dit is het beoordelen van een product door een expert. Dit is een benadering, die geen enkele aandacht aan de eis ‘reproduceerbaarheid’ besteedt. Het is immers de expert die het oordeel bepaalt. Deze wordt daarbij niet geleid door een meetinstrument. De expertbenadering komt in dit hoofdstuk niet verder aan de orde. Voor deze benadering zijn we geen methoden in de literatuur tegengekomen. Dit is wellicht te verklaren uit het feit dat juist de expert de beoordeling uitvoert: hij of zij zal hierbij wel

een stappenplan hanteren, maar dit stappenplan is persoonsafhankelijk. In hoofdstuk 6 komt de expertbenadering wel weer aan de orde. De drie benaderingen worden in de onderstaande figuur uitgebeeld op een schaal van subjectief naar objectief.



Figuur 3.4 Beoordelingsbenaderingen uitgezet op schaal subjectief versus objectief.

Bij het hiervoor geïntroduceerde onderscheid tussen codemetriek- en vraaggebaseerde beoordelingsbenadering speelt reproduceerbaarheid van de meting een belangrijke rol. In hoofdstuk 2 is aangegeven dat naast reproduceerbaarheid ook validiteit een eis aan metingen is. Bij de behandeling van validiteit is aangegeven dat er moet zijn gedefinieerd met welke metrieken een eigenschap wordt gemeten.

Codemetriek benaderingen gaan hierbij veelal uit van standaardmodellen: er wordt een model gehanteerd waarin gedefinieerd is welke metrieken welke kwaliteitseigenschap meten. Er wordt weinig tot geen aandacht besteed aan de vraag hoe men tot metrieken komt voor de betreffende eigenschappen.

Bij de vraaggebaseerde benadering wordt hier meer aandacht aan gegeven. In deze benadering ligt juist het accent op het bepalen van de kwaliteitskarakteristieken en vervolgens op het bepalen van de metrieken hiervoor. De vraaggebaseerde benaderingen leunen verder op intersubjectief meten, dat wil zeggen dat een oordeel tot stand komt door meer mensen te vragen naar hun oordeel en hen daarover overeenstemming te laten bereiken.

Naast het verschil in reproduceerbaarheid en aantonen van validiteit, speelt bij het onderscheid tussen de twee benaderingen ook het verschil in het onderwerp van beoordeling een rol. Het betreft hier het onderwerp van beoordeling (de scope) waarop de methode zich richt. De codemetriek gebaseerde methoden richten zich vrijwel alleen op de (source) code van het product, terwijl de vraaggebaseerde methoden zich naast de code ook richten op de documentatie van het product en soms zelfs op de organisatie rondom het product. De vraaggebaseerde methoden hebben daarmee een ‘bredere’ scope dan de codemetriek gebaseerde methoden.

Tabel 3.1 Vraag versus de codemetriek gebaseerde beoordelingsbenadering

	Codemetriek gebaseerd	Vraaggebaseerd
Reproduceerbaarheid	Sterk	Zwak
Aantonen validiteit?	Uitgaan van standaard modellen	Intersubjectief
Onderwerp van beoordeling	Code	Code, documentatie, proces

De hiervoor gepresenteerde tweedeling lijkt op de indeling die Bache en Bazzana (1994) geven voor beoordelingstechnieken (evaluation techniques). Deze indeling gaat uit van de wijze waarop een beoordeling wordt uitgevoerd. Dit kan door middel van statische analyse en dynamisch analyse, van inspectie en van modelleren. Als er wordt gemeten tijdens dynamische en statische analyses en bij modelleren dan is er meestal sprake van een codemetriek gebaseerde benadering. Bij inspectie is meestal sprake van de vraaggebaseerde beoordeling.

3.4 Codemetriek gebaseerde beoordelingsmethoden

3.4.1 Inleiding

De op codemetrieken gebaseerde beoordelingen maken gebruik van geautomatiseerde tools. Voorbeelden van dergelijke tools zijn:

- Cosmos – dit is een tool dat op de markt gebracht is door het Nederlandse bedrijf Techforce (tegenwoordig Emendo). Het oorspronkelijk breed inzetbare tool is de afgelopen jaren toegespitst op het testen van software op 'Jaar 2000 (Y2K) bestendigheid'. Deze versie van het tool heet Cosmos 2000. Het is onder andere toegepast in de 'Millennium Factory' van Cap Gemini (Emendo, 1999).
- Crocodile – een tool ontwikkeld aan de Universiteit van Cottbus (Duitsland). Het tool is geschikt voor analyses van object georiënteerde programmeertalen (Lewerentz en Simon, 1998).
- Datrix – een tool dat op de markt wordt gebracht door Bell Canada. Het wordt veel gebruikt op het Amerikaanse continent. Met het tool wordt statische analyse uitgevoerd (Daughtry e.a., 1990).
- Logiscope – een tool dat is ontwikkeld in het kader van het Scope-project. Het wordt op de markt gebracht door de Franse organisatie Verilog. Met het tool kan statische en dynamische analyse worden uitgevoerd op diverse programmeertalen (Verilog, 1992, 1998).
- Mood-kit – een tool dat voortkomt uit het in Portugal uitgevoerde Mood-project. Met het tool kan statische analyse worden uitgevoerd. Het is toegepast op met name object georiënteerde programmeertalen (Abreu e.a., 1994).
- QAC/C++ – een tool dat op de markt wordt gebracht door de Engelse organisatie Programming Research Ltd. Het is een statisch analysetool voor talen C en C++ (Programming Research Ltd, 1994).

We merken op dat de hierboven genoemde tools op zich nog geen beoordelingsmethoden hoeven te zijn. De tools zelf zijn compilers die een bepaalde set metrieken toepassen. Toch spreken we van beoordelingsmethoden, omdat ze vaak onderdeel van een stappenplan zijn. De betreffende beoordelingen kunnen dus worden opgevat als een methode, die wordt aangeduid met de naam van het onderliggend tool.

Van deze tools kiezen we één, namelijk Datrix, dat we verder analyseren. Er wordt gekozen voor Datrix omdat deze beoordelingsmethode uitgewerkt is in de literatuur (Mayrand and Coallier, 1996), (Laguë en April, 1996). De methode is in onze ogen representatief voor codemetriek gebaseerde beoordelingen. Daarnaast gaat het om niet zomaar een methode maar om een methode die reeds op grote schaal is en wordt toegepast. Datrix heeft in dat opzicht dus een bewezen staat van dienst.

3.4.2 Analyse van Datrix

Het Datrix-tool is van 1985 tot 1993 ontwikkeld aan de Ecole Polytechnique de Montreal. Tegenwoordig wordt het Datrix^{TM}-tool op de markt gebracht door Bell Canada. Deze tool bestaat uit de compiler om de structuur van de code vast te stellen, het tool om metrieken op deze structuur toe te passen en een database met ervaringsgegevens waarin aanbevelingen voor minima en maxima van metriekwaarden zijn opgenomen (Robillard e.a., 1991), (Daughtrey e.a., 1990).

Het Datrix tool is gebaseerd op statische analyse. Deze analyse is ‘the analysis of software products by other pieces of software (static analysers) where the software under examination is not actually executed’ (Bache and Bazzana, 1994). In plaats daarvan wordt de broncode als een geheel van controlestructuren: (sequentie, selectie en iteratie) beschouwd. De statische analyse bestaat eruit dat het tool de structuur van het programma nagaat. Bijvoorbeeld: hoeveel componenten zijn er en welke componenten roepen elkaar aan? Als het tool bijvoorbeeld nagaat welke componenten elkaar aanroepen, kan er een zogenaamde Call Graph –of flow graph- worden opgesteld. Dit is een grafische representatie van de samenhang tussen de componenten. Op basis van deze Call graph kunnen vervolgens een aantal metrieken worden berekend, bijvoorbeeld ‘number of edges’ en ‘number of nodes’. Voor een uitgebreide behandeling van statische analyse en call graphs wordt verwezen naar Dumke e.a. (1996) en Fenton Pfleeger (1996).

Beschrijving van de activiteiten

Eerste punt van de analyse is dat er wordt bekeken in hoeverre er bij Datrix sprake is van een beschrijving van activiteiten. Mayrand en Coallier (1996) geven hiervoor 10 stappen, namelijk:

1. Supplier agreement – het maken van afspraken tussen klant en beoordelaar over de manier waarop resultaten gebruikt mogen worden.

2. Product source code sample analysis and tool adjustment – het compileren van code op proef om te kijken of deze zonder problemen gecompileerd kan worden. Dit is van belang om de tools van de beoordelaar te testen op eventuele afwijkingen. Veel programmeertalen hebben een eigen 'dialect', waardoor de compiler van de beoordelaar in de problemen kan komen.
3. Programming and design guidelines: bepalen van de kwalificatieniveaus. Dit wordt gedaan door de beoordelaar die hiervoor design guidelines, waaraan de code volgens de klant moet voldoen, vertaalt in metrieken en measurement tresholds. Voorbeeld: als er single inheritance wordt vereist in een C++ product, dan dient de metric 'number of direct base class' niet de waarde 1 te overschrijden.
4. Source code static analysis – de product source files worden vertaald in files die metriekwaarden (lees: actuele meetwaarden) en graphs van deze code bevatten. Om deze stap te kunnen uitvoeren is het nodig dat de code compleet is en te compileren is.
5. Treshold utilization – het toepassen van de vooraf bepaalde tresholds. Treshold is in Datrix de term voor kwalificatieniveaus.
6. Treshold adjustment – hierbij wordt een database toegepast waarin meetwaarden van soortgelijke code uit eerdere beoordelingen zijn opgenomen. Een voorbeeld is de metriek 'Number of Paths' (NP) waarvan de maximum treshold op 10.000 wordt gesteld omdat uit ervaring kan worden afgeleid dat het tot 10.000 'static paths' nog steeds mogelijk is om een functie te begrijpen en een goed unit testing plan te definiëren.
7. Visual inspection – inspectie van de code om de resultaten van de metriek te bevestigen of te ontkrachten. Dit is een arbeidsintentieve en daarmee ´dure´ activiteit. Dit probleem wordt aangepakt door de inspectie te richten op de componenten die uit de analyse tijdens de voorgaande activiteit als zwak zijn gekwalificeerd. Mayrand en Coallier (1996) merken op dat bij deze inspecties soms naar voren komt dat metriekresultaten niet overeenkomen met inspectieresultaten. Zo werd bijvoorbeeld een bepaalde code door de metriek als complex gekwalificeerd, terwijl de programmeur de code als logisch en goed georganiseerd en daarmee gemakkelijk en begrijpelijk bestempelde.
8. architectural analysis – het bepalen van de ´macro-structuur´ (een totaalbeeld) van de code, bijvoorbeeld de afhankelijkheid tussen modules.
9. Assessment report – overleg met de klant over de resultaten. Deze kan eventueel commentaar leveren, waarmee vervolgens rekening gehouden wordt bij het opstellen van het eindrapport.
10. Multiversion software product tracking – er wordt nagegaan in hoeverre een volgende versie van het product verbeterd is. Hiervoor worden de metingen voor versie X gebruikt als een baseline voor de metingen van versie X+1. Op deze manier kan worden nagegaan wat er in de code veranderd wordt en of dit een verbetering is.

Eerder is aangegeven dat we tijdens de analyse de ISO-14598 activiteiten als referentie hanteren om de verschillende beoordelingsmethoden te evalueren. Als we dit voor de methode Datrix doen dan constateren we het volgende.

Datrix onderkent niet expliciet de ISO-activiteiten ‘bepaal doel van de beoordeling’ en ‘identificeer producttypen’. Daarentegen start Datrix met ‘supplier agreement’. In deze overeenkomst worden afspraken gemaakt tussen opdrachtgever en beoordelaar over welke resultaten een beoordeling oplevert en hoe deze resultaten te gebruiken. In deze zin betreft het een contract tussen opdrachtgever en opdrachtnemer. Hiermee ligt in elk geval impliciet vast welk product beoordeeld wordt en wat er met de beoordeling wordt beoogd. Daarom kunnen we de Datrix-activiteit ‘supplier agreement’ als equivalent beschouwen met de eerste twee ISO-activiteiten.

De volgende twee, door ISO 14598 onderkende, activiteiten zijn ‘specificeer kwaliteitsmodel’ en ‘specificeer metrieken’. Deze vinden we niet in Datrix terug. Dat is logisch omdat Datrix een vast stramien van beoordelen kent en beide ISO-activiteiten niet ter discussie stelt. Wie voor een Datrix-beoordeling kiest, weet hoe de het kwaliteitsmodel en de metrieken eruit zien.

Datrix kent wel de ISO-activiteit ‘bepalen kwalificatieniveaus voor metrieken’, zij het onder een andere naam, namelijk ‘programming and design guidelines’. In zekere zin gaat Datrix met deze activiteit in op het bepalen van het kwaliteitsmodel en de metrieken.

De ISO-activiteiten ‘bepalen criteria voor beoordeling’ en ‘formuleren van beoordelingsplan’ komen in Datrix weer niet expliciet aan de orde. Dit komt weer door het min of meer vaste stramien van het (geautomatiseerd) beoordelen, zodra de partijen elkaar gevonden hebben in de beoordelingsovereenkomst.

De ISO-activiteiten ‘uitvoeren van metingen’, ‘vergelijken van criteria’ en ‘vaststellen van resultaten; worden weer wel door Datrix onderkend, zij het onder andere namen, zie onderstaande tabel.

Datrix onderkent een aantal activiteiten die niet zo expliciet door ISO 14598 worden genoemd. Allereerst de activiteit ‘visuele inspectie’ en ‘architectural analysis’. Dit zijn Datrix-activiteiten die de geautomatiseerde meting aanvullen. Datrix beperkt zich daarmee niet tot de kwantitatieve metingen zoals codemetriek-gebaseerde methoden normaliter doen.

Tenslotte kent Datrix de activiteit ‘multi-version product tracking’ waarin wordt nagegaan in hoeverre nieuwere versies van de software verbeteringen zijn ten opzichte van de beoordeelde versie. Datrix onderkent hiermee dat er sprake is van opvolgende beoordelingen die elk een vervolg behoeven. De ISO-standaard spreekt zich op dit punt niet zo expliciet uit als de methode. De resultaten van voorgaande vergelijking worden samengevat in de volgende tabel.

Tabel 3.2 Vergelijking van activiteiten van ISO 14598 en Datrix-methode.

ISO 14598	Datrix
Bepaal doel van beoordeling	Supplier agreement
Identificeer producttypen	Supplier agreement
Specificeer kwaliteitsmodel	
Selecteer metrieken	
Bepalen van kwalificatieniveaus voor de metrieken	Programming and design guidelines
Bepalen criteria voor de beoordeling	
Formuleren van beoordelingsplan	
Uitvoeren van de metingen	Source code static analysis, visual inspection, architectural analysis
Vergelijken met criteria	Treshold utilization & treshold adjustment
Vaststellen van de resultaten	Assessment report
	Multiversion tracking

In bovenstaande vergelijking zijn twee Datrix-activiteiten niet aan de orde gekomen, namelijk 'product source code sample analysis and tool adjustment' en 'source code static analysis'. Beide activiteiten zijn specifiek voor een codemetriek gebaseerde methode. Als zodanig zijn ze geen onderwerp in de veel meer als 'open' te karakteriseren ISO-activiteitenlijst.

Als we terugkijken op de analyse dan constateren we dat er ondanks enkele verschillen toch een grote mate van overeenkomst tussen de Datrix- en ISO 14598-activiteiten is. Uiteraard verschilt de naamgeving en zijn er verschillen in accenten wat betreft de concrete invulling van de activiteiten. Maar in hoofdlijnen geldt voor beide benaderingen hetzelfde stramien.

Processtructuur

Bij het aspect ‘processtructuur’ is nagegaan in hoeverre de methode relaties onderkent tussen de verschillende activiteiten. hierop gaat de methode slechts weinig in: de activiteiten worden los van elkaar beschreven. Er zijn echter wel een aantal onderlinge afhankelijkheden tussen de activiteiten die de methode niet expliciet aangeeft, maar die tussen de regels door zijn te achterhalen. Zo kent de methode als tweede activiteit: 'product source sample analysis'. Dit betreft een proef om van de te beoordelen software na te gaan of deze compileerbaar is. Hiermee wordt onderkend dat er kan worden teruggekoppeld op de activiteit 'supplier agreement'.

Aansturen van het proces

De Datrix-methode geeft ook weinig aandacht aan de aansturing van het proces. Wat betreft de inrichting van het proces gaat de methode uit van een vast volgorde van de activiteiten. Er wordt niet expliciet onderkend dat er alternatieve volgordes mogelijk zijn. Wel komen we impliciet soms aanwijzingen voor de inrichting tegen. Zo kan men de afhankelijkheid tussen de activiteiten 'product source sample analysis' en 'supplier agreement', beschreven bij het

voorgaande aspect, opvatten als een onderwerp van de inrichting van het proces. Deze relatie wordt echter zo impliciet aangeduid dat er niet echt sprake is van inrichting.

De methode geeft wel aandacht aan de toewijzing van de middelen. Per activiteit geeft de methode aan welke soort mensen –ontwikkelaar, opdrachtgever, eindgebruiker– betrokken dient te zijn bij een activiteit. Ook wordt de inzet van het tool gedefinieerd per activiteit.

De voortgangsbewaking van het proces komt in de methode niet aan de orde. Er vindt geen tracking en tracing op de uitgevoerde activiteiten plaats.

Dit relatief gesloten karakter van de methode ligt voor de hand omdat met het toepassen van Datrix duidelijk voor ogen staat dat er code op een geautomatiseerde wijze moet worden beoordeeld. Er is daarmee een vast stramien. Pas in de laatste stappen veranderd dit als Datrix activiteiten onderkent als 'visual inspection', 'architecture analysis'. Bij deze activiteiten gaat het voornamelijk om overleg tussen opdrachtgever en de uitvoerder. Daarbij is veel meer interactie en daarmee is meer sturing nodig.

Afwegen van doel en middelen

De constatering ten aanzien van beide voorgaande aspecten geldt ook voor het aspect afwegen van doen en middelen. De methode onderkent amper enige vorm van afweging tijdens het proces. Ze veronderstelt dat alle vereiste middelen, zowel in kwantiteit als in kwaliteit, beschikbaar zijn. Voor het eerste deel van het proces is dit wellicht terecht. Voor het laatste deel van het proces geldt dit zeker niet. Hier zal men wel degelijk te maken krijgen met de vraag tot in welke mate van detail en daarmee met welke inzet van middelen activiteiten uitgevoerd moeten worden. Dit geldt bijvoorbeeld ten aanzien van de visuele inspectie. Bij deze activiteiten zal de balanceerproblematiek zeker aan de orde komen. De methode zegt daar echter weinig over.

Terugkoppelen en bijstellen van het proces

Met betrekking tot dit aspect scoort Datrix beduidend beter dan wat betreft de drie voorgaande aspecten. In de methode zitten verschillende aandachtspunten voor terugkoppeling. De eerste terugkoppeling is voorzien vanuit de tweede activiteit 'product source sample analysis'. Bij het subaspect 'inrichting van proces' is al aangegeven dat de resultaten van deze activiteit worden teruggekoppeld op de voorgaande activiteit: 'supplier agreement'. Deze terugkoppeling leidt in het meest negatieve geval tot de conclusie dat de beoordeling niet volgens de overeenkomst kan worden uitgevoerd. Minder radicaal is de bijstelling dat er afspraken worden gemaakt over doorlooptijd en budget, omdat er bijvoorbeeld geïnvesteerd moet worden in het ‘compileerbaar maken’ van de code. Beide voorbeelden hebben te maken met het bijstellen van doel en ambitieniveau.

Een ander moment van terugkoppeling komt aan de orde bij de activiteiten 'treshold utilization' en 'treshold adjustment'. Datrix beveelt aan om de meetresultaten uit geautomatiseerde activiteiten terug te koppelen naar de ontwikkelaars. Hierbij neemt de

methode aan dat de personen goed een oordeel over de code en daarmee over de meetresultaten kunnen geven. Deze terugkoppeling acht de methode noodzakelijk om de zogenaamd objectieve meting correct te kunnen inspecteren. Dit kan betekenen dat uit de geautomatiseerde procedure naar voren komt dat bepaalde meetwaarden niet tussen de normwaarden liggen. Echter, als ontwikkelaars goede redenen hebben voor dergelijke afwijkingen dan zullen dergelijke afwijkingen (veelal) worden geaccepteerd. We constateren dat Datrix wel degelijk aandacht besteedt aan terugkoppeling en bijstellen. In die zin wordt voldaan aan het vijfde aspect van ons analysekader.

3.5 Vraaggebaseerde beoordelingsmethoden

3.5.1 Inleiding

Er bestaan vele beoordelingsmethoden die we tot de vraaggebaseerde benadering kunnen rekenen. Onderstaande opsomming geeft hiervan een beeld:

- Scope– een Esprit onderzoeksproject dat heeft geresulteerd in een beoordelingsmethode (Robert, 1994), (Qiu, 1995), (Bache and Bazzana, 1994).
- MicroScope – methode die wordt toegepast door de Deense instelling Delta. MicroScope komt voort uit Scope (Andersen, 1993), (Kyster, 1995).
- Q-Seal – methode die wordt toegepast door het Italiaanse Q-Seal-consortium waaronder Etnoteam. Methode komt voort uit Scope (Caliman, 1996).
- Afotec – methode toegepast binnen US Airforce om de onderhoudbaarheid van hun softwaresystemen te bepalen (Afotec, 1996).
- TAQS – een Braziliaans onderzoeksproject dat heeft geresulteerd in een beoordelingsmethode. Voor uiteenlopende producten wordt bepaald of deze producten aan de ISO 9126 kwaliteitskarakteristieken voldoen. Beoordelingen resulteren in het uitroepen van een product tot ‘best software of the year’ in de Braziliaanse software industrie (Tsukumo e.a., 1996).
- Maintainability Index – methode, ontwikkeld door de Britse CCTA om de onderhoudbaarheid van uiteenlopende informatiesystemen te bepalen (West, 1994). Een andere opzet voor een Maintainability Index wordt beschreven in (Stark en Oman, 1995), (Welker e.a., 1997).
- Software Portability Assessment Method (SPAM) – een Europees onderzoeksproject dat heeft geresulteerd in een methode om portabiliteit van een softwareproduct te bepalen (Omi/Spam consortium, 1997).
- Lloyd Register’s Software Conformity Assessment – een methode om te bepalen of software voldoet aan de ISO 12119 standaard. De methode wordt uitgevoerd door de Engelse organisatie Lloyd Register’s die ook een certificaat afgeven (Lloyd´s Register, 1994).

Het valt buiten de scope van dit proefschrift om alle methoden te analyseren. Er wordt daarom volstaan met de behandeling van 2 van de opgesomde methoden, namelijk Scope en Afotec. Scope wordt behandeld omdat het van grote invloed is geweest op het denken over softwareproductbeoordelingen. Zo vinden een aantal belangrijke publicaties over beoordelen hun oorsprong in het Scope-project (Rae e.a., 1995), (Bazzana en Bache, 1994), (Hausen and Welzel, 1993). Ook de procesbeschrijving van ISO 14598 (zie paragraaf 3.2) vindt zijn oorsprong in het Scope-project.

Scope is feitelijk een project dat heeft geleid tot een aanpak die wij opvatten als een echte beoordelingsmethode. In het verlengde van het Scope-project hebben een aantal van de Scope-partners zelf beoordelingsdiensten ontplooid, zoals: MicroScope in Denemarken (Kyster, 1995), Q-Seal (Caliman, 1996) en Ibisco (Maiocchi, 1997) in Italië. Ook het van 1996 tot 1998 uitgevoerde Space-Ufo (1998) project is sterk geïnspireerd door de aanpak van het Scope-project. Tijdens het project is een methode voor het opstellen van kwaliteitsprofielen ontwikkelt (Space-Ufo consortium, 1998). Zie voor kwaliteitsprofielen hieronder.

Naast Scope besteden we ook aandacht aan Afotec. Deze methode is representatief voor vraaggebaseerde methoden, die zich richten op het beoordelen van slechts één bepaalde kwaliteitskarakteristiek. Dit in afwijking van de Scope-methode, waarbij het vaststellen van kwaliteitskarakteristieken onderdeel van de methode is.

3.5.2 Analyse van Scope

Scope staat voor Software CertificatiOn Programme in Europe. Het Scope-project (Esprit, 2nd Framework, 2151) is uitgevoerd van 1989 tot 1993 (Robert, 1994). Het had tot doel om van beoordelingstechnieken te ontwikkelen en toe te passen voor softwareproducten.

Beschrijving activiteiten

Hiervoor is al aangegeven dat het Scope-project van grote invloed is geweest op de ISO 14598 standaard. De beschrijvingen van activiteiten lijken dan ook sterk op elkaar. Een verschil is dat Scope zich voornamelijk richt op het niveau van wat ISO 14598 als deelprocesen aanduidt. De Scope-methode omschrijft wel wat er binnen de verschillende deelprocessen kan gebeuren, maar dit is niet zo gedetailleerd als in ISO 14598. Het onderstaande geeft de activiteiten zoals die binnen Scope zijn onderkend door Robert (1994). Boegh e.a. (1992) geven een vergelijkbare uitwerking.

– analyse evaluation requirements – bepalen van de eisen die aan de beoordeling worden gesteld. Eisen worden geformuleerd in termen van te beoordelen karakteristieken van het product en in termen van beoordelingsniveaus,

- specification of evaluation – bepalen van de te beoordelen onderdelen van het product en nadere aanscherping van de kwaliteitskarakteristieken voor het product en de beoordelingsniveaus,
- design of evaluation – bepalen van de te gebruiken beoordelingstechniek. Hierbij wordt zoveel mogelijk gebruik gemaakt van standaard evaluatiemodules,
- performing the evaluation – uitvoeren van de beoordeling. Dit wordt door Scope aangeduid als inspectie, meten of testen. Tijdens dit deelproces worden de in de vorige stap gekozen evaluatiemodules gebruikt,
- reporting on the result – rapportage van de bevindingen naar de opdrachtgever.

In de volgende tabel worden voorgaande activiteiten vergeleken met de door ISO 14598 gedefinieerde activiteiten.

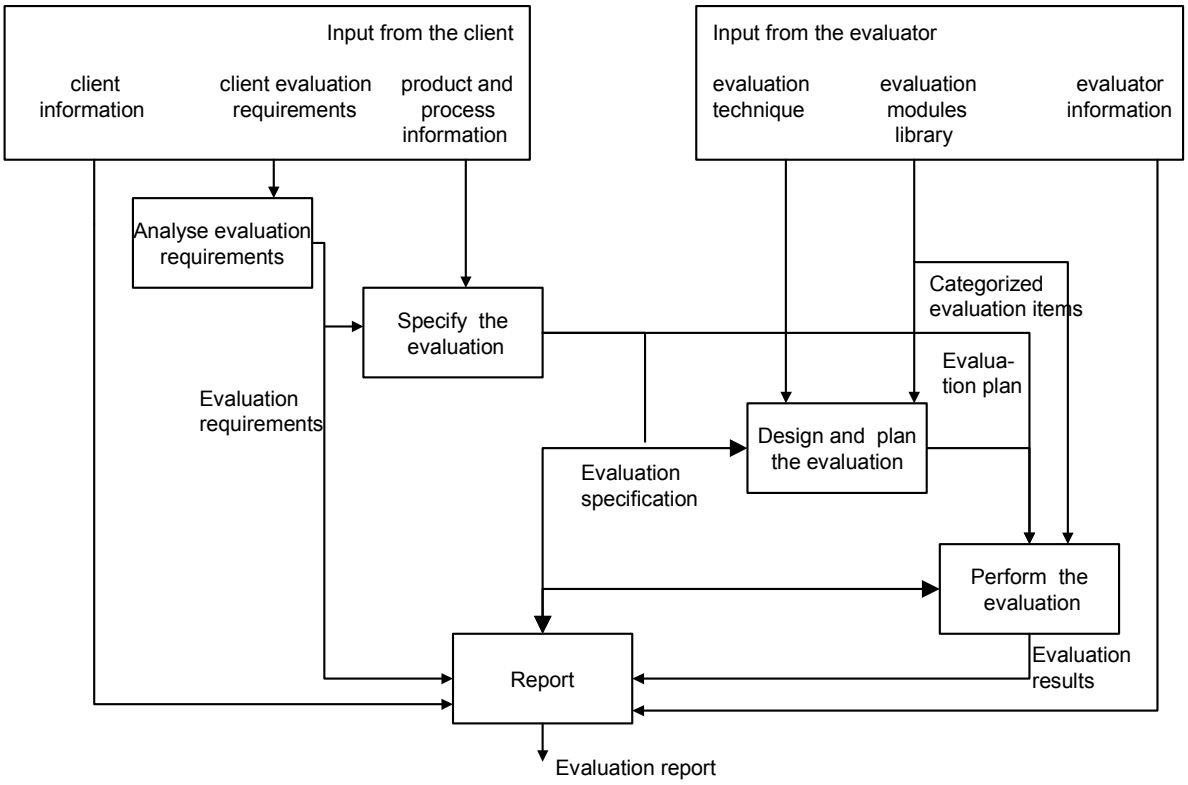
Tabel 3.3 Vergelijking van activiteiten van ISO 14598 en Scope-methode

ISO 14598	Scope-methode
Bepaal doel van beoordeling	
Identificeer producttypen	Specification of evaluation
Specificeer kwaliteitsmodel	Analyse evaluation requirements
Selecteer metrieken	Specification of evaluation
Bepalen van kwalificatieniveaus voor de metrieken	Specification of evaluation
Bepalen criteria voor de beoordeling	
Formuleren van beoordelingsplan	Design of evaluation
Uitvoeren van de metingen	Performing the evaluation
Vergelijken met criteria	
Vaststellen van de resultaten	Reporting on the result

De activiteiten van Scope komen in grote mate overeen met die van ISO 14598. Dat is ook logisch omdat de ISO-standaard voor een belangrijk deel voortkomt uit te resultaten van het Scope project. In die zin is ISO een verdere uitwerking en de detaillering van Scope. Bij Scope blijft echter het onderscheid tussen ‘Analyse evaluation requirements’ en ‘Specification of evaluation’ onduidelijk. De bedenkers van Scope hebben hier wellicht mee beoogd om eerst een kritische analyse uit te voeren, alvorens te kiezen voor bepaald kwaliteitsmodel en de daarbij passende metrieken. Op zich is er best wat te zeggen voor deze aanpak. Men wordt zo immers gedwongen om eerst goed te analyseren wat men wil beoordelen en hoe dat zou kunnen gebeuren, voordat men tot de definitieve beoordelingsaanpak besluit.

Processtructuur

Met betrekking tot het aspect processtructuur onderkent de Scope-methode allerlei relaties tussen de activiteiten. Deze worden in de volgende figuur schematisch uitgedrukt.



Figuur 3.5 Processtructuur volgens Scope (Robert, 1994)

Voorgaande figuur geeft aan hoe de diverse activiteiten samenhangen en in welke volgorde ze worden uitgevoerd. Ook zijn de informatiestromen tussen de activiteiten en van en naar de omgeving (opdrachtgever, beoordelaar) beschreven. Scope benoemt deze informatiestromen tussen de activiteiten en werkt ze kort uit. Op deze manier onderscheidt Scope zich positief ten opzichte van veel andere methoden, waar de processtructuur en informatieoverdracht veel minder aandacht krijgen

Aansturen van het proces

Eerder is de aangegeven dat bij aansturing drie zaken van belang zijn, namelijk: het inrichten van het beoordelingsproces, het toewijzen van middelen aan de activiteiten en het bewaken van de voortgang. Wat betreft inrichting geeft Scope wel degelijk aanknopingspunten en richtlijnen, zie hiervoor. Wat betreft de middelenallocatie en voortgangsbewaking zegt de methode vrijwel niets. Blijkbaar gaat de methode ervan uit dat dit geregeld is en dat het niet in de methode behandelt hoeft te worden.

Afwegen van doel en middelen

De Scope-methode onderkent expliciet de noodzaak tot het afwegen van doel en middelen. Middelen betreffen onder andere de zogenaamde beoordelingsmodules (Eng. Evaluation module). Dit zijn hulpmiddelen voor de activiteiten ‘uitvoeren meten’ en ‘vergelijken met de criteria’ (ISO 14598 terminologie). Deze beoordelingsmodules worden beschouwd als de

'bouwstenen' van een beoordeling. In het begin van het Scope-project werden ze dan ook aangeduid als 'bricks' (Scope consortium, 1992a, 1992b). Dit concept is later overgenomen in de ISO 14598-standaard (ISO FDIS 14598-6, 1999). De modules worden geordend op basis van de kwaliteitskarakteristieken uit het ISO 9126 kwaliteitsmodel en beoordelingsniveaus. Elke module bevat een verzameling metrieken. De metrieken bestaan veelal uit vragen met gesloten antwoordmogelijkheden. Het relatieve belang van een karakteristiek wordt uitgedrukt middels het toekennen van het beoordelingsniveau. Het concept beoordelingsniveau is gebaseerd op het idee dat er in sommige gevallen meer aandacht aan de waarde van een karakteristiek moet worden gegeven dan in andere gevallen. Er worden vier niveaus onderkend: A (hoog) tot D (laag). Het concept beoordelingsniveau hangt samen met het begrip 'Software Integrity Level' uit ISO 61508 (1998).

Tijdens het Scope-project zijn maar liefst 96 beoordelingsmodules ontwikkeld (Scope consortium, 1992b, 1993a), (Qiu, 1995). Deze modules zijn alle bruikbaar voor de beoordelingsniveaus D en C. Voor de niveaus A en B zijn vooralsnog geen modules ontwikkeld. Onderstaande tabel geeft een overzicht van de tijdens het project ontwikkelde modules.

Tabel 3.4 Overzicht van beoordelingsmodules per kwaliteitskarakteristiek

ISO 9126 karakteristiek	aantal opgestelde beoordelingsmodules
Functionaliteit	56
Betrouwbaarheid	1
Bruikbaarheid	12
Onderhoudbaarheid	25
Efficiëntie	-
Portabiliteit	2

Bij de afweging van doel en middelen speelt bij de Scope-methode expliciet de vraag op welk beoordelingsniveau men een bepaalde kwaliteitskarakteristiek wil meten. Uiteraard hangt dit samen met de investering in tijd, geld, mensen. Immers hoe hoger het beoordelingsniveau hoe zwaarder de beoordeling, maar ook hoe duurder. Keuzes van kwaliteitsniveau en kwaliteitskarakteristieken kan men vastleggen in een kwaliteitsprofiel In dit profiel zijn de relevante karakteristieken vastgelegd met de bijbehorende beoordelingsniveaus. Onderstaande figuur geeft een voorbeeld van een ingevuld profiel.

	D	C	B	A
Functionality		x		
Usability			x	
Reliability		x		
Efficiency		x		
Maintainability	X			
Portability	X			

Figuur 3.6 Voorbeeld van een kwaliteitsprofiel

Een methode voor het opstellen van dergelijke kwaliteitsprofielen is opgesteld binnen KEMA Nederland en TU Eindhoven (Heemstra e.a., 1994), (Van der Zwan, 1995), (Eisinga e.a., 1995) en tijdens het Space-Ufo-project (Space-Ufo consortium, 1998), (Van Ekris, 1998).

Terugkoppelen en bijstellen
Bij Scope is geen sprake van terugkoppeling. Het beoordelingsproces is als het ware 'plat': het loopt van links naar rechts. Elk deelproces heeft opvolgers, maar er wordt niet teruggekoppeld naar voorgangers. Bij gebrek aan terugkoppeling zal men bij Scope dan ook weinig tot niets aantreffen over bijstellen. Beide zaken liggen in elkaars verlengde.

3.5.3 Analyse van Afotec

Afotec staat voor Air Force Operational Test and Evaluation Center. Met de methode wordt geprobeerd om de Maintainability van softwaresystemen binnen de Amerikaanse luchtmacht te bepalen. De in de Afotec omschreven doelstelling is: 'to measure and report a maintainability rating to help resolve a measure of effectiveness for software supportability and to identify deficiencies to the acquirer and developer to facilitate product improvements and to help the organization responsible for post-deployment software support' (Afotec, 1996).

De methode is onderdeel van een bredere aanpak –de zogenaamde Operational Test and Evaluation (Afotec, 1996). De Afotec methode zelf beschrijft hoe een Maintainability beoordeling te plannen, uit te voeren en er over te rapporteren. De beoordeling wordt uitgevoerd op een informatiesysteem als dit systeem 'production representative' is. Dit betekent dat de code en documentatie vrijwel af zijn en er geen belangrijke wijzigingen meer worden verwacht.

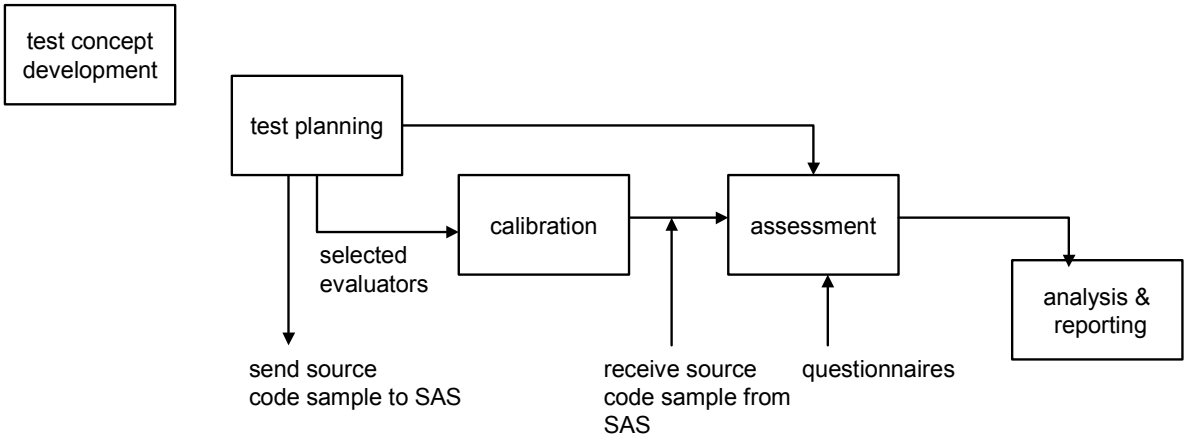
Beschrijving van de activiteiten
Het beoordelingsproces onderkent vijf fasen, die we opvatten als activiteiten:

1. Test concept development – het opstellen van een plan waarin wordt aangegeven welke beoordeling wordt uitgevoerd, welke software beoordeeld wordt, wanneer het wordt beoordeeld en welke middelen nodig zijn om de beoordeling uit te voeren.
2. Evaluation planning – het formeren van een beoordelingsteam, het verzamelen van beoordelingsmateriaal, vaststellen van program/module hiërarchie en het op basis hiervan bepalen van steekproeven op de broncode. Er worden steekproeven genomen omdat het beoordelen van een compleet systeem veel werk is en daardoor te duur zou zijn. De steekproeven worden in principe centraal, via het zogenaamde instituut SAS, onderzocht.
3. Calibration – beoordelaars worden getraind in de achtergrond en betekenis van de vragen. Hierdoor moeten de beoordelaars op zelfde kennisniveau worden gebracht, waarna ze zoveel als mogelijk dezelfde normen hebben om de vragen te beantwoorden.
4. Assessment – uitvoeren van de beoordeling door de beoordelaars. Tijdens een beoordeling voorziet een beoordelaar alle items van een waarde die vervolgens worden ingevoerd in een tool. Dit tool genereert per beoordelaar een score en een totaalscore over alle beoordelaars van het systeem.
5. Analysis and reporting – analyse van de resultaten door een moderator, gevolgd door een rapportage.

Het heeft weinig zin om de activiteiten van Afotec te vergelijken met die van ISO 14598. Afotec is immers een methode die volledig is toegespitst op het beoordelen van de karakteristiek Maintainability. ISO 14598 is daarentegen bedoeld voor beoordelingen waarin verschillende karakteristieken aan de orde kunnen komen. Vandaar dat er activiteiten als ' bepaal doel van beoordeling' en 'specificeer kwaliteitsmodel' worden behandeld. Bij Afotec zijn dergelijke activiteiten overbodig omdat reeds vaststaat dat er beoordeeld wordt op Maintainability. Het bijzondere van de Afotec-methode is dat er expliciet aandacht besteed wordt aan calibratie. Wanneer men verschillende beoordelaars bij een beoordeling betrekt, dan is vooraf niet gegarandeerd dat zij allen dezelfde normen hanteren met dezelfde strengheid; zie het onderwerp reproduceerbaarheid in hoofdstuk 2. Het is dan ook zaak de beoordelaars in deze op elkaar af te stemmen, te calibreren. Afotec doet dit door middel van trainingen, waardoor de beoordelaars dezelfde kennis over de vragen hebben. Hierdoor hoopt men dat de beantwoording zo uniform mogelijk, met zo weinig mogelijk variantie, geschiedt. Aan de activiteit 'calibratie' geeft de ISO 14598 nauwelijks aandacht

Processtructuur

De Afotec-methode geeft het volgende schema waarin beschreven staat hoe de activiteiten onderling samenhangen.



Figuur 3.7 Processtructuur in Afotec (1996)

De figuur laat zien dat de activiteit 'test concept development' los staat van de andere activiteiten. Dit wijst op een onderbroken keten. Voor de andere activiteiten beschrijft de methode redelijk gedetailleerd welke informatie nodig is voor het uitvoeren van de activiteiten en welke vervolgactiviteiten voortbouwen op de uitvoer van voorgaande activiteiten.

Aansturen van het proces
De methode gaat uit van een volgorde van het uitvoeren van activiteiten zoals hiervoor onderkent. De methode onderkent niet dat er andere volgorde(s) gevolgd kunnen worden. De methode besteedt aandacht aan de middelentoewijzing, zo geeft de methode aan welke personen –welke functies binnen de luchtmacht organisatie- bepaalde activiteiten dienen uit te voeren. Ook is per activiteit gedefinieerd welke tools moeten worden gebruikt. De methode gaat niet expliciet in op de voortgangsbewaking.

Afwegen van doel en middelen
In de methode komt de afweging van doel en middelen slechts zijdelings aan de orde. Tijdens de eerste activiteit 'test concept development' wordt bepaald wat de in te zetten tijd en inspanning is. Het gaat dan om de inzet van mensen, niet om de inzet van technieken. De methode beschrijft welke personen nodig zijn per stadium van de beoordeling. Het doel verandert bij die afweging niet. Er wordt uitgegaan van een vast omschreven doel, namelijk: 'het bepalen van de Maintainability van het te beoordelen systeem'. Wel kan de omvang van de steekproeven van het te beoordelen product worden gewijzigd. Men balanceert dus met name de middelen op basis van de capaciteit waarmee ze worden ingezet.

Terugkoppelen en bijstellen
De processtructuur beschreven in Figuur 3.7 laat zien dat er geen sprake is van expliciete terugkoppelingen tijdens het proces. De uitvoer van een activiteit wordt alleen door de opvolgende activiteit gebruikt, er worden geen activiteiten bijgesteld.

Dat er geen aandacht is voor terugkoppeling wordt bevestigd door een survey uitgevoerd door Baumann (1996). Tijdens deze survey ondervroeg hij diverse organisaties over het toepassen van Afotec beoordelingen. 42 US Airforce organisaties werden aangeschreven in de periode september 1995 tot januari 1996, waarbij 24 organisaties reageerden. De totale survey omvatte 51 vragen. Een deel van zijn survey richt zich op het bepalen van de tevredenheid over de methode. Hiervoor stelde hij onder andere de in dit verband relevante vraag 'compared to the current Afotec method, how should a new maintainability evaluation method differ?' De respondenten hadden de keuze uit 8 antwoordmogelijkheden die hieronder staan vermeld. Respondenten mochten maximaal 3 punten toekennen aan alle 8 mogelijkheden, waarbij de belangrijkste 2 zou krijgen, de middelste 1 punt en de minst belangrijke 0 punten. De resultaten in de onderstaande tabel zijn zo genormaliseerd dat een score 8.0 betekent dat alle respondenten dit punt als het belangrijkste aangaven.

Tabel 3.5 Antwoorden op de vraag uit de survey van Baumann (1996)

time – evaluation should take less that one week	1.1
effort – evaluation should require fewer than 5 evaluators	1.6
scope – evaluation should cover more than 10% of system software	3.0
accuracy – evaluation scores should match more accurately system's true maintainability	4.1
feedback –faster turn-around between completion evaluation and report issuing	5.2
realistic – evaluation should be more operationally realistic	6.2
support – greater use of tools to perform repetitive tasks	3.9
size – questionnaire should contain fewer questions	2.2

De tabel laat zien dat de onderdelen 'feedback', 'realistic' en 'accuracy' als het meest belangrijk worden ervaren. Het punt 'accuracy' wijst op de discrepantie tussen het door Afotec gemeten begrip Maintainability en de opinie van engineers hierover. Baumann concludeert zelf dat 'maintainability evaluation in general is viewed as a useful process, but the results of past evaluations are not being used to their full potential'. De uitkomsten van deze survey laten zien dat het ontbreken van terugkoppeling als een probleem wordt ervaren door de Afotec gebruikers.

Ten aanzien van terugkoppelen naar het ambitieniveau constateren we dat de Afotec-methode hieraan geen aandacht besteed. Dat de gebruikers van de methode wel behoefte aan deze terugkoppeling hebben wordt bevestigd door het volgende resultaat uit Baumann's survey. Op de vraag 'How important is it to you to have an Afotec evaluation that yields a score that is compatible with your existing or planned maintainability metric?' geeft het grootste deel van de respondenten (16 personen) aan dat de metriek betrekking zou moeten hebben op verschillende ambitieniveaus, met name gerelateerd aan kosten, doorlooptijd en benodigde menskracht. Dit duidt erop dat de betreffende personen niet steeds hetzelfde doel voor ogen hebben bij een beoordeling en verschillende ambitieniveaus onderkennen.

3.6 Interpretatie en probleemdefinitie

3.6.1 Evaluatie van de analyse van beoordelingsmethoden

Dit hoofdstuk heeft tot doel om de in de literatuur beschreven beoordelingsmethoden te analyseren om problemen ten aanzien van het beoordelingsproces uit te werken. Hiervoor is het analysekader opgesteld waarin de nadruk is gelegd op de besturing van een beoordelingsproces. Door systematische afleiding van de in de systeemtheorie beschreven 'voorwaarden voor effectieve besturing' zijn we uitgekomen op een kader waarin vijf aspecten een rol spelen. Dit analysekader is vervolgens gehanteerd bij het bespreken en becommentariëren van beoordelingsmethoden. In deze paragraaf blikken we terug op deze analyse om vervolgens te komen tot een aanscherping van de probleemstelling van dit onderzoek.

Beschrijving van activiteiten

Het eerste aspect van het analysekader betreft de vraag of een methode een lijst van activiteiten onderkent, met andere woorden: wordt er gedefinieerd wat onder het beoordelingsproces wordt verstaan? Als we de resultaten van de voorgaande paragrafen bekijken, dan stellen we vast dat alle besproken methoden aandacht aan dit aspect besteden. Elke methode kent een lijst van activiteiten, zij het dat in veel gevallen zowel aard als benaming van de activiteiten verschilt. Tijdens de analyse is de activiteitenlijst van ISO 14598 als referentiekader gebruikt. Het bleek dat geen van de besproken methoden deze standaard volledig gebruikt. In die zin is er nog lang geen sprake van een algemeen geaccepteerde 'de facto' standaard voor een beoordelingsproces. Dit was ook niet te verwachten gezien de recente publicatiedatum van de ISO 14598: in 1998 en 1999. De besproken methoden stammen uit de tijd voor deze datum.

Wat betreft de codemetriek en de vraaggebaseerde beoordelingen valt op dat er tussen beide soorten accentverschillen bestaan in activiteiten. In het algemeen hebben de vraaggebaseerde methoden een meer open karakter. Er is niet op voorhand vastgelegd welk soort product zal worden beoordeeld, naar welke karakteristiek hierbij wordt gekeken en met welke metrieken wordt gemeten. Al deze zaken staan bij veel van de vraaggebaseerde beoordelingen open. Tijdens het beoordelingsproces moet hierover overeenstemming bereikt worden. De Afotec-methode is op dit punt een uitzondering, omdat deze methode zich richt op een bepaalde kwaliteitskarakteristiek, Maintainability, en daardoor een meer gesloten karakter heeft.

De codemetriek gebaseerde beoordelingen hebben sowieso een meer gesloten karakter. In deze methoden is het kwaliteitsmodel bij wijze van spreken ‘ingebakken’. Hierover wordt niet meer gediscussieerd. Een activiteit zoals ‘specificeer kwaliteitsmodel’ wordt in dergelijke methoden dan ook niet aangetroffen.

Het voorgaande heeft gevolgen voor het aspect ‘aansturen van het proces’. Immers één van de subaspecten van aansturen betreft de inrichting: welke activiteiten zijn nodig in een beoordelingsproces, lettend op de situationele kenmerken. Dit punt komt bij het aspect ‘aansturing’ uitvoeriger aan de orde.

Processtructuur
Eén van de voorwaarden voor effectieve besturing is dat een besturend orgaan een model moet hebben van het bestuurde systeem. Zo'n model omvat niet alleen een precieze omschrijving van uit te voeren activiteiten, maar ook inzicht in en overzicht over de samenhang van deze activiteiten.

Als we de diverse methoden uit de literatuur op dit punt bekijken, dan valt op dat de meeste methoden slechts weinig zeggen over die processtructuur. Een positieve uitzondering hierop is de Scope-methode, die ingaat op de processtructuur door onder andere de informatiestromen te benoemen tussen diverse activiteiten. Daarmee wordt duidelijk welke in- en uitvoerafhankelijkheden er bestaan en dus ook in welke volgorde men de activiteiten dient uit te voeren.

Aansturen van het proces
Zoals eerder is toegelicht omvat het aansturen van een proces op z'n minst drie deelactiviteiten weten:
- procesinrichting: welke activiteiten worden uitgevoerd, gegeven de specifieke context en in welke volgorde wordt dit gedaan,
- middelenallocatie: toewijzen van vereiste middelen aan de diverse activiteiten en
- voortgangsbewaking: nagaan of de uitvoering van activiteiten volgens plan verloopt.

Slechts weinig methoden besteden expliciet aandacht aan alle subaspecten. Daarmee is aansturing nauwelijks een onderwerp. Blijkbaar gaan de methoden ervan uit dat dit zich zelf wel regelt tijdens de uitvoering van het proces.

Met betrekking tot de inrichting van het proces verwachten we op z’n minst, dat de methoden daarvoor richtlijnen en adviezen geven. Het beoordelen van software is immers geen rechttoe, rechtaan proces. Voordat men een beoordelingsproces kan uitvoeren dient men zich eerst kritisch af te vragen welke situationele factoren er spelen. Vervolgens worden op basis daarvan keuzes gemaakt zoals: welke activiteiten wel of niet nodig zijn, tot in welke mate van detail en in welke volgorde. Voorbeelden van situationele factoren zijn: het type product dat wordt beoordeeld, het ambitieniveau van de beoordeling, de daaruit voortvloeiende mate van nauwkeurigheid en detail, de rol en invloed van betrokken partijen en de beschikbare middelen in opzicht van zowel kwaliteit als capaciteit. Afhankelijk van deze factoren wordt het proces ingericht.

Aan het subaspect ‘toewijzen van middelen’ wordt door verschillende methoden aandacht besteed. Het gaat dan om het toewijzen van zowel mensen als technieken. Zo onderkennen verschillende methoden –Scope en Afotec- welke mensen met welke functie per activiteit moeten worden ingezet. Ook wordt meestal wel aangegeven van welke technieken gebruikt moeten worden gemaakt in een bepaalde activiteit.

Het subaspect ‘voortgangsbewaking’ komt vrijwel niet aan de orde. Geen van de methoden onderkent een mogelijkheid of de noodzaak om na te gaan of alle activiteiten worden uitgevoerd.

Afwegen van doel en middelen
Ten aanzien van het afwegen van doel en middelen geldt een soortgelijke conclusie als bij het aansturen van het proces. Vrijwel geen enkele methode besteedt expliciet aandacht aan de noodzaak om doel en middelen tegen elkaar af te wegen. Hoofdredenen daarvan zijn het ontbreken van een expliciete stap doelformulering en onvoldoende inzicht in vereiste c.q. beschikbare middelen. Beide punten worden hieronder kort uitgewerkt.

Onvoldoende doelformulering houdt in dat methoden wellicht onderkennen dat er een doel moet worden opgesteld, maar dat ze vervolgens niet aangeven hoe dat kan gebeuren en op welke aspecten daarbij te letten. De Scope-methode en het vervolg daarop, de Space-Ufo-methode, zijn eigenlijk de enige die daar hier een uitzondering op vormen. Deze methoden onderkennen namelijk een kwaliteitsprofiel. Dit profiel legt vast welke karakteristieken van een product moeten worden beoordeeld en tevens op welk beoordelingsniveau dat moet gebeuren. Daarmee wordt de eerste stap gezet richting afweging van doel en middelen.

Andere methoden besteden ofwel geen enkele aandacht aan doelformulering of gaan ervan uit dat een doel weliswaar wordt vastgesteld, maar dat er over zo'n doel niet onderhandeld wordt. Een doel is dan een vastomlijnd gegeven waaraan niet getornd kan worden. Vanzelfsprekend is vanuit dit perspectief geen enkele afweging van doel en middelen mogelijk.

Een tweede hoofdreden van het gebrek aan doel-middel afwegingen, is dat methoden vrijwel geen oog hebben voor het afwegen zelf. Als er over de inzet van mensen en technieken wordt gesproken dan heeft dit betrekking op het subaspect toewijzen van middelen (zie voorgaande aspect), niet zozeer op het afwegen van deze middelen op het doel van de beoordeling. Een uitzondering daarop is de Scope-methode. Deze methode gaat expliciet in op het afwegen door expliciet te onderkennen dat er verschillende evaluatiemodules voor een bepaald kwaliteitsprofiel –in feite de doelformulering- kunnen worden geselecteerd. Hiermee geeft de methode expliciet ruimte aan de afwegingsdiscussie.

De Scope-methode gaat hiermee in op het afwegen van technieken. Ten aanzien van de afweging met betrekking tot mensen benodigd voor het uitvoeren van de beoordeling, zegt

geen van de methoden iets; ook niet wanneer sprake is van experts. Er wordt blijkbaar verondersteld dat menskracht nooit een probleem is, nog qua kwantiteit, nog qua kwaliteit. De praktijk leert dat dit wel degelijk een probleem vormt en dat het daarom zo belangrijk is om expliciet doel-middel afwegingen te maken.

Terugkoppelen en bijstellen van proces
De beoordelingsmethoden besteden vrijwel geen aandacht aan het terugkoppelen van resultaten tijdens het proces en aan het terugkoppelen van het eindresultaat op het gestelde doel. Een en ander impliceert dat ook het bijstellen van het proces erbij inschiet. Er zijn slechts enkele uitzonderingen die de regel bevestigen. Zo beveelt de Datrix-methode aan om de meetresultaten die verkregen zijn met behulp van het meettool terug te koppelen op de ontwikkelaars van de software. Dit doet men omdat dergelijke resultaten niet contextvrij zijn en dat er daarom interpretatie door de ontwikkelaars nodig is, waarna eventueel een vervolgmeting moet worden uitgevoerd (bijstelling).

Van terugkoppeling van het eindresultaat naar de doelstelling, gevolgd door het eventueel bijstellen van doel en ambitieniveau is in geen enkele methode sprake. Hier wreekt zich opnieuw de aanname dat een doel een vaststaand iets is, waaraan niet getornd wordt; zie het aspect afweging doel en middelen. De methoden gaan in dit verband uit van rationeel handelende partijen. Dat het beoordelen van een softwareproduct ook, in sommige gevallen zelfs met name, een politiek onderhandelingsproces is, wordt door geen van de methoden onderkent. Zij veronderstellen een rationeel proces waarbij gestreefd wordt naar een optimale beoordeling, in plaats van een beoordeling waarin sprake is van een ‘satisficing’ strategie. Dit laatste wil zeggen dat men zo goed als mogelijk recht doet aan de verschillende belangen van de betrokken belanghebbenden.

3.6.2 Een aangescherpte probleemdefinitie

Dit proefschrift is begonnen met de constatering dat softwareproducten steeds belangrijker worden in onze samenleving. Daarmee neemt ook het belang van de kwaliteit van dergelijke producten toe. Er is echter geconstateerd dat er onvrede bestaat over de kwaliteitsbeoordelingen van software. Dit leidde in hoofdstuk 1 tot de probleemstelling dat het echte manco van deze beoordelingen ligt in de onvoldoende besturing van beoordelingsprocessen. De analyse van de in de literatuur verschenen beoordelingsmethoden bevestigt deze probleemstelling. Bij geen van de methoden is sprake van effectieve besturing van het proces. De problematiek rondom besturing is hiermee toegespitst op drie aspecten, namelijk:

- aansturen van het proces,
- afwegen van doel en middelen,
- terugkoppelen en bijstellen

Met het ontbreken van de afweging van doel en middelen is naar voren gekomen dat ook het formuleren van een doel een belangrijk onderwerp van besturing is en dat dit door de methoden veelal wordt onderschat.

Het voorgaande leidt tot de volgende aangescherpte probleemstelling: het beoordelen van softwareproducten leidt veelal tot onbevredigende resultaten, doordat het ontbreekt aan een goede besturing van het beoordelingsproces. Tekortkomingen in deze besturing liggen met name op de volgende vier terreinen:

- er is onvoldoende aandacht voor het formuleren van een operationele doelstelling, zodanig dat alle betrokken partijen zich daarin herkennen,
- door onder andere onvoldoende inzicht in de activiteiten van het beoordelingsproces en de samenhang daartussen (processtructuur) schort het aan een goede aansturing van beoordelingsprocessen (inrichten, toewijzen van middelen, bewaken van de voortgang),
- besturing houdt ook in dat doel en middelen worden afgewogen. Bestaande beoordelingsmethoden onderkennen dit balanceervraagstuk nauwelijks, maar veronderstellen 'vaste' doelen en een oneindige capaciteit aan middelen, zowel in kwantitatief als kwalitatief opzicht,
- essentieel voor het beoordelingsproces zijn het terugkoppelen van resultaten, eventueel gevolgd door het bijstellen van doel, middelen en activiteiten. Geen van de in de literatuur beschreven methoden voldoet aan dit aspect van besturing.

Deze probleemstelling is getoetst tijdens het uitvoeren van twee casestudies in de praktijk. Deze praktijktoets bevestigde dat de genoemde 4 deelproblemen uiterst relevant zijn. Ze zijn daarom uitgangspunt voor ons ontwerp ter verbetering van softwareproductbeoordelingen. Voordat dit ontwerp in hoofdstuk 6 aan de orde komt, worden eerst de resultaten van de casestudies in hoofdstuk 4 en 5 gepresenteerd.

4. Omega casestudie

Het doel van deze casestudie is om aan te geven dat de vier problemen die zijn gesteld in hoofdstuk 4, relevante problemen zijn in de beoordelingspraktijk. In de casestudie is het proces geanalyseerd van het beoordelen van het Omega 2050 systeem bij Tokheim RPS.

4.1 Verantwoording

Deze casestudie betreft de beoordelingsmethode van het Omega 2050-systeem. Het Omega-systeem en de achtergronden bij de beoordeling ervan worden toegelicht in paragraaf 4.2. De beoordeling van het systeem wordt uitgevoerd met behulp van codemetrieken. Dit onderwerp wordt uitgewerkt in paragraaf 4.3. Vervolgens komen de ervaringen met de beoordeling aan de orde. In paragraaf 4.4 wordt beschreven dat betrokkenen bij de beoordeling –en dan met name de ontwikkelaars van het Omega-systeem- ontevreden over de beoordeling zijn. Deze ontevredenheid vormt het uitgangspunt van de casestudie.

Rol van de onderzoeker

Het feit dat er ontevredenheid bestond over de beoordeling bij de ontwikkelaar van het Omega 2050 systeem, deed de behoefte aan advies ontstaan bij de organisatie. Het betrof advies over de toe te passen codemetrieken. De vragen van de organisatie waren:

1. Worden de juiste metrieken toegepast om de Maintainability van Omega-modules te bepalen?
2. Worden de normwaarden voor deze metrieken goed bepaald?

Deze adviesopdracht is uitgevoerd door de onderzoeker in samenwerking met de Quality Engineer van RPS en specialisten van KEMA Nederland B.V. (Punter, 1998b), (Punter, 1999). De opdracht is deels uitgevoerd onder auspiciën van het Space-Ufo-project (Space-Ufo consortium, 1998). Het resultaat was een advies waarin aanvullende metrieken zijn voorgesteld. Ook is een voorstel voor andere normwaarden gedaan.

Parallel aan de adviesopdracht werd een casestudie uitgevoerd. Deze casestudie richtte zich op de analyse van het proces van beoordelen. Er werd gebruik gemaakt van het analysekader uit hoofdstuk 3. In het kader van de casestudie werden betrokkenen geïnterviewd en is interne documentatie bestudeerd. Zo werd bepaald of de vier –in hoofdstuk 3 bepaalde– problemen zich in de beoordelingsprocessen voordeden. Vanuit het toegepaste analysekader zijn verbeteringsvoorstellen gedaan aan de organisatie en is de analyse van de metrieken gebruikt tijdens de casestudie.

Aanpak casestudie en informatiebronnen
De casestudie is gestart met het analyseren van de ontevredenheid van de ontwikkelaars over de beoordeling. Hiervoor is een experiment uitgevoerd waarin de resultaten van de bestaande Omega-beoordeling zijn vergeleken met een voor de gelegenheid gecreëerde beoordeling door ontwikkelaars. Het waren immers de ontwikkelaars die ontevreden over de beoordeling waren. De resultaten van dit experiment worden in paragraaf 4.4 gepresenteerd.

Na het experiment is vanuit het analysekader naar vier aspecten van de besturingsproblematiek van beoordelen gekeken, namelijk:
- Doelformulering – hoe wordt het doel opgesteld?
- Aansturing van proces – welke activiteiten worden beschreven en hoe hangen ze samen (processtructuur)?
- Afwegen van doel en middelen – hoe worden de middelen geselecteerd, welke afwegingen worden daarbij gemaakt?
- Terugkoppeling en bijsturing – hoe wordt (eventuele) terugkoppeling uitgevoerd?

Deze aspecten zijn onderzocht aan de hand van observaties van de onderzoeker. Hiervoor zijn in totaal zes interviews met betrokken partijen (Quality Engineer, ontwikkelaars) georganiseerd. Verder is relevante documentatie aangaande de ontwikkeling van de Omega-beoordelingsmethode bestudeerd (Revai, 1997), (Spirits, 1997). Tijdens het uitvoeren van de adviesopdracht zijn ook een aantal opmerkingen van betrokkenen genoteerd en vervolgens geïnterpreteerd in termen van het analysekader. Tenslotte is gekeken naar andere codemetriek gebaseerde benaderingen om de werkwijze tijdens de Omega-beoordelingsmethode te plaatsen. De resultaten van deze analyse worden in de paragrafen 4.5 tot en met 4.8 gepresenteerd.

4.2 Omega-systeem

Het Omega 2050 systeem is een verkoop informatiesysteem dat gebruikt wordt op benzinestations. Het ondersteunt de managers en operators niet alleen bij de verkoop van olieproducten, maar ook voeding- en luxeartikelen die tegenwoordig op benzinestations verkocht worden. Het Omega-systeem verzorgt de afhandeling bij verkoop, voorraadregistratie en signaleert als er artikelen opnieuw besteld moeten worden. Het systeem kan modulair worden uitgebreid. De meest eenvoudige versie betreft de afhandeling van een pomp met een aantal verkoopfuncties. De meest uitgebreide versie is een volledig verkoopstation met een backoffice functie, te vergelijken met de functionaliteit van een supermarkt.

Het Omega 2050-systeem wordt geproduceerd en op de markt gebracht door Retail Petroleum Systems (RPS). Dit is een onderdeel van Tokheim, een bedrijf dat marktleider is op het gebied van 'retail petroleum products' en diensten daarvoor. Tijdens het uitvoeren van

de casestudies was de RPS-divisie onderdeel van Schlumberger. De divisie is in september 1998 verkocht aan Tokheim.

Oorspronkelijk werd het Omega systeem ontwikkeld en onderhouden door de ontwikkelafdeling. Deze afdeling is gevestigd op twee locaties, namelijk in Bladel (Nederland) en in Monrouge (Frankrijk). De oorspronkelijke ontwikkelstrategie ging ervan uit dat deze centrale afdeling een basissysteem aflevert aan de lokale werkmaatschappijen van Tokheim. Deze operational companies (Opco's) zijn vervolgens verantwoordelijk voor het afstemmen van het systeem per land of regio. Dit houdt in dat het basissysteem zo wordt ingesteld dat het voldoet aan de wetgeving en specifieke eisen die elk Europees land aan dergelijke verkoopstations stelt. Zo mag er in Nederland al getankt worden op een pomp als de voorgaande tankende automobilist nog niet heeft afgerekend. In Duitsland is dit niet toegestaan: daar moet een automobilist eerst afrekenen voordat de volgende automobilist mag gaan tanken.

Over deze oorspronkelijke ontwikkelaanpak van het Omega-systeem bestond ontevredenheid bij de Opco's. Deze werkmaatschappijen vonden dat zij nieuwe versies van het basissysteem opgelegd kregen en dan maar moesten zorgen dat zij deze versies aanpasten aan de eisen die elke lokale markt stelt. Vanuit de Opco's was er behoefte om zelf het ontwikkelproces aan te sturen. De Opco's hebben uiteindelijk hun zin gekregen: er is een nieuwe ontwikkelstrategie gelanceerd. Deze aanpak houdt in dat de Opco’s de centrale ontwikkelafdeling aansturen. Hierbij signaleren de Opco’s ieder de voor hun markt specifieke trends. Deze vertalen ze in gewenste nieuwe functionaliteit voor het Omega-systeem. Deze wensen van iedere Opco afzonderlijk worden in een groter verband met de andere Opco’s besproken. Als andere Opco’s deze functionaliteit ook wensen, dan krijgt de ontwikkelafdeling opdracht dit te implementeren in het basissysteem. Dit betekent veelal dat de bij één Opco ontwikkelde softwaremodule wordt opgenomen in het basissysteem, waarna ook de andere Opco's er over kunnen beschikken.

Door deze nieuwe ontwikkelstrategie ontstond bij de centrale ontwikkelafdeling de behoefte om de kwaliteit van de aangeboden software modules te bepalen voordat de modules in het basissysteem worden opgenomen. De ontwikkelaars kenden de aangeboden modules veelal niet en wilden daarom zekerheid over de kwaliteit ervan. Dit was het begin van de Omega-beoordelingsmethode.

4.3 Omega-beoordelingsmethode

Oorspronkelijk bestond de beoordeling eruit dat de ‘coding standard conformance’ van de modules werd bepaald. Dit hield in dat de code van de modules werd beoordeeld op basis van een aantal regels die zijn opgesteld in de codeerstandaarden. Deze codingstandards werden door ontwikkelaars toegepast bij het programmeren in C en C++. Het betrof bijvoorbeeld het voorschrift dat de regels code niet langer dan 80 karakters mogen zijn. De

codeerstandaarden werden in de loop der jaren door de centrale ontwikkelafdeling opgesteld (Bruyninkx e.a., 1993). Voor de beoordeling zijn de regels geïmplementeerd in een tool. Op deze manier kan van de modules geautomatiseerd worden vastgesteld of ze voldoen aan de codeerstandaarden. Hierbij wordt gebruik gemaakt van de tools CodeCheck en PCLint. Het resultaat van deze beoordeling is een rapport waarin per module is aangegeven wat de afwijkingen ten opzichte van de standaard zijn.

Het ontwikkelteam was echter niet tevreden over deze manier van beoordelen. De ontevredenheid betrof de rapporten waarin per module de afwijkingen van de standaard worden gepresenteerd. Men kon veelal geen oordeel vellen over de kwaliteit van de module. Men vond het moeilijk om op basis van de geconstateerde afwijkingen te beslissen of het de moeite waard was om de module verder in ontwikkeling te nemen of dat deze moeite niet lonend was en dat de module beter opnieuw ontwikkeld kon worden. Dit was de aanleiding voor het uitbreiden van een beoordeling die zich beperkte tot check op coding standard conformance naar een beoordeling waarin de kwaliteit van de modules wordt bepaald. Onder kwaliteit werd de onderhoudbaarheid (Maintainability) verstaan. Als een module onderhoudbaar is dan loont het de moeite om de module te blijven aanpassen. De achterliggende motivering hiervoor was: als een module niet onderhoudbaar wordt bevonden, dan kan de module beter opnieuw ontwikkeld worden.

Deze herziene vorm van de Omega-beoordeling wordt aangeduid als kwaliteit- of Maintainability-beoordeling. Voor deze kwaliteitsbeoordeling werd de oorspronkelijk opzet uitgebreid tot een beoordeling waarin metrieken worden toegepast. De uitkomsten van deze metingen moeten een antwoord geven op de vraag of modules onderhoudbaar zijn of niet.

De Omega kwaliteitsbeoordeling maakte gebruik van metrieken. Het betrof codemetrieken, 15 stuks in totaal. Voorbeelden waren: SumOfLinesOfCode en NumberOfHighLevelStatements. Met deze metrieken werden de actuele waarden bepaald. Deze werden vervolgens vergeleken met de normwaarden. Deze normwaarden betroffen de minimum- en maximumwaarden waartussen een actuele meetwaarde dient te liggen. Het begrip normwaarde is synoniem met het begrip kwalificatieniveau uit ISO 14598. De normwaarden voor de Omega-beoordelingsmethode zijn afgeleid uit het Logiscope-referentiemodel. De volgende tabel geeft een aantal van deze metrieken en de daarvoor vastgestelde normwaarden.

Tabel 4.1 Enkele metrieken uit de kwaliteitsbeoordeling en hun normwaarden

Naam van de metriek	Afkorting	Min	Max
SumOfLinesOfCode	SumOfLoC	0	60
NumberOfHighLevelStatements	SumOfHighStatm	0	2
AverageStatementSize	AvgOfStatmSize	1	9
Vocabulary frequency /	AvgOfVocFreq	0,5	5
Nested level count /	SumOfNestLevels	0	4

De actuele meetwaarden werden bepaald met behulp van een statisch analysetool. De vergelijking van deze waarden met de normwaarden vindt geautomatiseerd plaats. De statische analyse en de vergelijking –met bijbehorende databases– zijn geïntegreerd in één tool dat wordt aangeduid als Anac (Spirits, 1997). De gebruikers van het tool kregen na meting en interpretatie een overzicht van het aantal functies van een module dat voldoet aan de eisen. Hierbij werden de functies van een module tot de volgende categorieën geclassificeerd: 'uitstekend', 'goed', 'acceptabel' en 'slecht'. Deze categorieën gaven de mate aan waarin de module onderhoudbaar was, ofwel: voldeed aan de karakteristiek Maintainability. De volgende figuur is een voorbeeld van wat een gebruiker van het tool na het meten en interpreteren te zien kreeg.

Module	Uitstekend	Goed	Acceptabel	Slecht
Module 1	0,00%	60,44%	39,56%	0,00%
Module 2	0,00%	44,44%	50,00%	5,56%
Module 3	0,00%	100,00%	0,00%	0,00%
Module 4	16,00%	64,00%	20,00%	0,00%

Figuur 4.1 Resultaten bepaald door het beoordelingstool voor vier Omega modules

De meetresultaten in bovenstaande figuur laten bijvoorbeeld zien dat 60,44% van de functies van module 1 'goed' voldeed aan de karakteristiek Maintainability. Omega-modules werden geaccepteerd als ze niet scoorden in de kolom 'slecht'. Voor de modules in bovenstaande tabel betekent dit dat module 2 werd afgewezen.

4.4 Tevredenheid over de beoordeling

De kwaliteitsbeoordeling is eind 1997 voor het eerst uitgevoerd. De aanpak is gebruikt om veertien Omega-modules te beoordelen. Eind 1998, een jaar na de invoering, was de kwaliteitsbeoordeling echter niet meer in gebruik.

Een belangrijke reden hiervoor was de ontevredenheid bij de ontwikkelaars over de uitgevoerde beoordeling. De ontwikkelaars kwalificeerden de metrieken die werden toegepast tijdens de beoordeling als een slechte voorspeller van de kwaliteit van de modules. Ontwikkelaars meldden een aantal keer dat ze het oneens waren met het door de metrieken berekende oordeel over Omega-modules. Zo kwam het voor dat de module volgens de tijdens de kwaliteitsbeoordeling toegepaste metrieken geaccepteerd moest worden, terwijl achteraf bleek dat er beter opnieuw ontwikkeld had kunnen worden. Ook de omgekeerde situatie kwam voor: een module werd in eerste instantie afgewezen door het tool, maar uiteindelijk toch opgenomen omdat achteraf bleek dat er voldoende bruikbare elementen aanwezig waren.

Een complicerende factor hierbij was de introductie van een nieuw ontwikkelingsplatform binnen RPS. Het tool dat de kwaliteitsbeoordeling ondersteunde sloot niet goed op dit platform aan. Dit resulteerde erin dat gegevens uit de ontwikkelingomgeving handmatig moesten worden overgezet naar het tool om zo de metrieken te berekenen. Deze complicatie leidde samen met de ontevredenheid over de resultaten tot het aan de kant schuiven van het tool. De investering in het tool om deze aan te laten sluiten op de nieuwe ontwikkelomgeving had een lage prioriteit binnen de organisatie.

Hiervoor is al aangegeven dat de ontevredenheid van de ontwikkelaars over de beoordeling, aanleiding was voor het uitvoeren van de casestudie. In deze paragraaf worden de resultaten van een experiment gepresenteerd waarin we inzicht probeerden te krijgen in deze ontevredenheid. Tijdens dit experiment zijn de resultaten van de kwaliteitsbeoordeling vergeleken met de resultaten van een beoordeling door de ontwikkelaars. We zijn er hierbij vanuit gegaan dat als beide beoordelingen vergelijkbaar scoren, de kwaliteitsbeoordeling niet zo slecht is als de kritiek erop ons wil doen geloven.

Het experiment is uitgevoerd door de metriekwaarden en de meningen van ontwikkelaars over dezelfde Omega modules met elkaar te vergelijken. Dit werd gedaan om te bepalen of de metriekwaarden en de meningen vergelijkbare resultaten opleveren. Onder vergelijkbaar verstaan we een verband tussen het aantal actuele meetwaarden dat tussen de normwaarden ligt en de hoogte van de score door de ontwikkelaars. We verwachtten dat naarmate het aantal actuele meetwaarden tussen de normwaarden per module groter is, ook de gemiddelde ontwikkelaarscore per module hoger is.

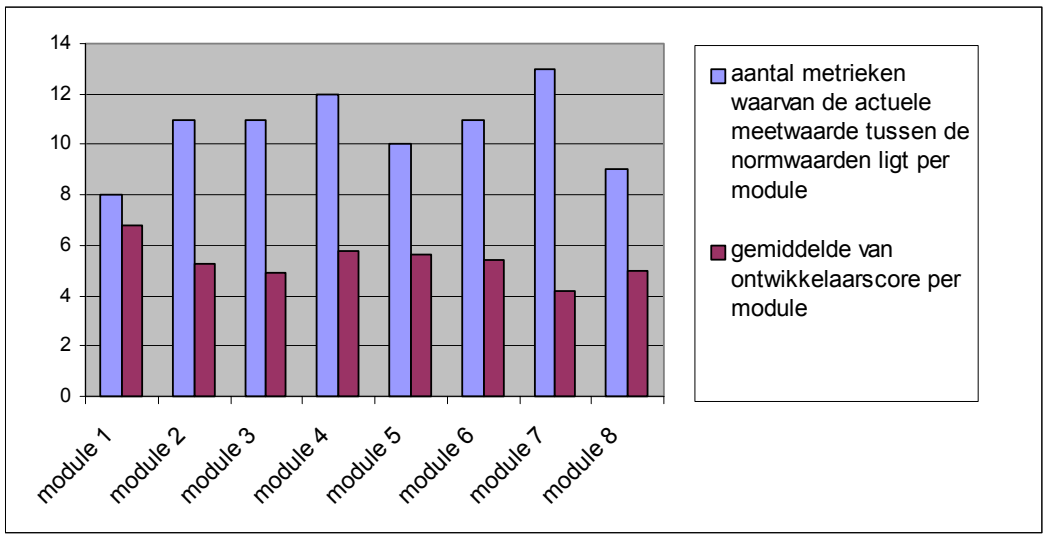
De ontwikkelaars is gevraagd om hun oordeel over de onderhoudbaarheid van de module te bepalen en op papier te noteren. Ook konden ze hun reden voor hun oordeel op dit papier kwijt. De opinie van de ontwikkelaars is gescoord op een schaal van 1 tot 10. Deze schaal moet net als het rapportcijfer op een Nederlandse school worden geïnterpreteerd: 1 is zwaar onvoldoende, 10 uitstekend en 5,5 (afgerond: 6) is de grens tussen onvoldoende en voldoende. Voor het experiment is gebruik gemaakt van de meningen van 5 ontwikkelaars.

Elke ontwikkelaar had ervaring met het onderhoud van de betreffende modules. Hierbij wordt opgemerkt dat de omstandigheden bij het uitvoeren van het experiment daarmee anders zijn dan in het geval van een reguliere kwaliteitsbeoordeling van Omega-modules. Tijdens een reguliere kwaliteitsbeoordeling hebben de ontwikkelaars geen ervaring met de modules: de modules worden dan immers door de Opco aangeboden zonder dat de ontwikkelaars ervaring met de module hebben. Tijdens het experiment hadden de ontwikkelaars wel ervaring met het onderhoud van de modules.

Tijdens het experiment is gemeten aan 8 Omega-modules. Voor elke module zijn de meetwaarden en ontwikkelaarsscores bepaald. Vervolgens is het aantal actuele meetwaarden

tussen de normwaarden per module bepaald, alsook de gemiddelde score van de 5 ontwikkelaarscores.

De volgende figuur presenteert de resultaten van beide metingen. De cijfers zijn geïndexeerd. Dit om beide soorten gegevens met elkaar te vergelijken. De ontwikkelaarscore wordt immers gemeten op een 0 tot 10 schaal, terwijl het aantal metriekenwaarden dat tussen de normwaarden valt, is bepaald op een 0 tot 15 schaal.



Figuur 4.2 Aantal metrieken waarvan meetwaarde tussen de normwaarden ligt per module vergeleken met gemiddelde ontwikkelaarscore per module.

De figuur laat niet een overduidelijke relatie tussen beide variabelen zien. Zo werd module 1 door de ontwikkelaars als beste module bepaald, terwijl module 7 als slechtste module werd aangeduid. Echter, bij module 1 is het aantal metrieken dat tussen de normwaarden ligt beduidend lager dan bij module 7. De laatst genoemde module heeft zelfs het grootste aantal metriekwaarden tussen de normwaarden vergeleken met de andere modules.

Als we de modules 1 en 7 buiten beschouwing laten, door ze als ‘outliers’ (Fenton en Pfleeger, 1996) te bestempelen, dan vinden we voor de overige modules wel steeds eenzelfde verhouding tussen beide variabelen. Het aantal metrieken waarvan meetwaarde tussen de normwaarden ligt is steeds, ongeveer, twee keer zo groot als de gemiddelde engineerscore. Daarmee lijkt er een verband te bestaan tussen beide beoordelingen: een hogere score door ontwikkelaars is gerelateerd aan een groter aantal metrieken waarbij de meetwaarde tussen de normwaarden ligt. Hiermee wijkt de kwaliteitsbeoordeling niet veel af van de lijn die de meningen van de ontwikkelaars representeert.

Op basis van de beperkte dataset (gegevens over 8 modules) kunnen we geen vérstrekkende conclusies aan dit experiment verbinden. Zo is er geen sprake van een statistisch verband.

Het experiment is ook niet bedoeld om aan te tonen dat beide beoordelingen overeenkomen. Wel laat het zien dat de meeste uitkomsten van de kwaliteitsbeoordeling redelijk overeenkomen met de meningen van de ontwikkelaars. De ontevredenheid van de ontwikkelaars maakt echter duidelijk dat er nadere analyse nodig is. Dit wordt gedaan door te bekijken in hoeverre de vier problemen die in hoofdstuk 3 ten aanzien van de besturing van het beoordelingsproces zijn opgesteld aan de orde komen. De resultaten van deze analyse komen in de volgende paragrafen aan de orde.

4.5 Ervaringen met het formuleren van het doel van beoordeling

Hiervoor zijn al een aantal doelformuleringen gepresenteerd. Zo is het oorspronkelijke doel van de beoordeling aangeduid; deze luidde: ‘bepaal of de code voldoet aan de standaard: aantonen coding standard conformance’. Dit doel veranderde met de overgang naar de kwaliteitsbeoordeling. Toen werd het doel geformuleerd als: ‘bepalen van Maintainability van code’.

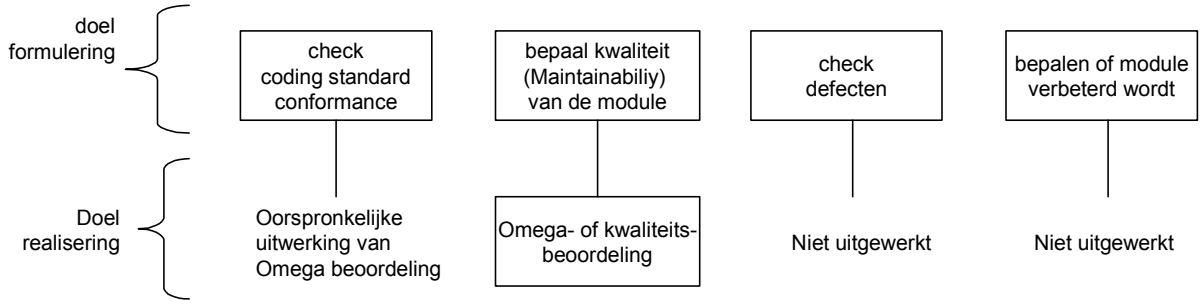
Doelformulering in termen van kwaliteitskarakteristiek

De formulering van het doel rondom de kwaliteitsbeoordeling luidde: ‘kwaliteit van de modules bepalen om te beslissen over het accepteren of afwijzen van modules’. Deze formulering is zo algemeen, dat het op veel manieren kon worden geoperationaliseerd. Het doel werd dan ook al snel concreter verwoord. Dit werd gedaan door het in termen van kwaliteitskarakteristieken (Maintainability) te formuleren. Deze kwaliteitskarakteristiek is daarmee onderdeel van de doelformulering.

In voorgaande doelformulering werd het onderwerp van de beoordeling aangeduid als module. Hieronder kan van alles worden verstaan. Naast code, kan er bijvoorbeeld ook documentatie onder worden begrepen. Ook kan men kijken naar zaken als interface met gebruiker, de aan te roepen subprogramma’s, enzovoorts. In de uitwerking van de beoordeling is de nadruk van de beoordeling komen te liggen op broncode. Kennelijk was er behoefte om het onderwerp van beoordeling explicieter te omschrijven, dan de bestaande formulering die de term module hanteert.

Het beschrijven van het doel in termen van kwaliteitskarakteristiek en onderwerp van beoordeling leek in eerste instantie op het mixen van twee verschillende zaken. De karakteristieken verwoorden immers de eisen die aan het product worden gesteld, dit is in principe wat anders dan de formulering van een doel. Toch bleek het voor de organisatie nuttig om karakteristiek en onderwerp van beoordeling te gebruiken om de doelstelling explicieter te beschrijven.

Discrepantie tussen doel en uitwerking door veranderende doelen
Op basis van gesprekken die met de ontwikkelaars en kwaliteitsmanagement zijn gevoerd, zijn nog andere dan het hiervoor aangehaalde doel opgetekend, namelijk: ‘bepaal het aantal fouten in de modules’ en ‘bepaal of de code beter wordt gedurende het onderhoud’. Deze doelen zijn in de loop der tijd gaan leven bij de mensen en worden naast de ‘officiële’ onderkende doelstelling nagestreefd. Het probleem hierbij is dat alleen de formulering ‘bepaal kwaliteit van beoordeling’ wordt uitgewerkt met de huidige kwaliteitsbeoordeling. De andere doelen vereisen een eigen uitwerking die niet is gemaakt. De daarmee ontstane discrepantie tussen doel en uitwerking wordt geïllustreerd in de volgende figuur.



Figuur 4.3 Verschillende doelformulering, maar één uitwerking.

De twee doelen die rechts in de figuur worden afgebeeld zijn onderkend tijdens de casestudie, maar zijn dus niet uitgewerkt, terwijl de ontwikkelaars deze doelen wel op de Omega-(kwaliteits)beoordeling projecteren. Dit leidt tot onvrede bij ontwikkelaars omdat deze doelen niet kunnen worden nagestreefd met de huidige uitwerking van de beoordeling.

Interpretatie
We constateren dat de doelformulering onvoldoende is. Door de weinig expliciete formulering van het oorspronkelijke doel, zijn andere doelen geformuleerd. Deze doelen worden echter niet allemaal gedekt door de huidige uitwerking van de beoordeling. Hierdoor kan er ontevredenheid over de beoordeling ontstaan: in de beoordeling herkent men niet het na te streven doel.

Ten aanzien van de beschreven doelformuleringen valt op dat ze alle genoemd werden door ontwikkelaars van het Omega-systeem. De afnemers van het systeem, de Opco’s, hebben het doel van beoordeling niet meebepaald. Deze partij is buiten de beoordeling gehouden door de ontwikkelaars. Dit verklaren we uit het streven van de ontwikkelaars om de beoordeling ‘in eigen hand te houden’.

4.6 Ervaringen met het aansturen van het proces

Het aspect aansturen van proces is in het analysekader van hoofdstuk 3 uitgewerkt middels drie subaspecten, namelijk: inrichting van proces, toewijzen van middelen en voortgangsbewaking.

Als we kijken naar de inrichting van het proces dan stellen we vast dat het proces vrijwel niet is gedefinieerd. Alleen de activiteit 'uitvoering' is beschreven. Achteraf constateren we echter dat er meer activiteiten zijn uitgevoerd, namelijk:

- selecteren metrieken,
- vaststellen van de normwaarden,
- vaststellen van de criteria,
- uitvoering van de beoordeling,

Selecteren metrieken – tijdens deze activiteit werd vastgesteld welke metrieken tijdens de kwaliteitsbeoordeling werden gebruikt. De metrieken werden gekozen op basis van het feit dat ze in het Logiscope referentiemodel worden genoemd als metrieken om Maintainability te meten. Er werden 15 geselecteerd, waarvan in tabel 4.1 een aantal voorbeelden zijn opgesomd.

Vaststellen normwaarden – na de metriekselectie werden voor elk van de metrieken normwaarden bepaald. De normwaarden werden afgeleid uit het Logiscope referentiemodel (Verilog, 1998), (Visser, 1997).

Vaststellen criteria – tijdens deze activiteit werd bepaald hoe van de geïnterpreteerde meetwaarden tot een uitspraak over het al dan niet voldoen aan Maintainability te komen. Hiervoor werd een model opgesteld waarin werd aangegeven hoe een meetwaarde die aan de normwaarden voldoet te vertalen naar een uitspraak over de subkarakteristiek. Vervolgens werden de waarden geaggregeerd zodat een uitspraak over Maintainability mogelijk werd.

Uitvoering – de metingen werden uitgevoerd door de code te analyseren met behulp van een statisch analysetool. De resultaten van deze analyse werden automatisch geïnterpreteerd met behulp van hetzelfde tool.

Elk van de Omega-activiteiten is afgebeeld op de door ISO 14598 onderkende activiteiten; zie hoofdstuk 3. De activiteit 'selecteren metrieken' betrof hetzelfde als de gelijknamige ISO-activiteit. Vandaar dat de Omega-activiteit op de ISO-activiteit wordt afgebeeld. De Omega-activiteit 'vaststellen normwaarden' betrof de ISO 14598 activiteit 'bepalen van kwalificatieniveaus voor de metrieken'. De Omega-activiteit 'uitvoering' betrof twee ISO 14598-activiteiten, namelijk zowel het 'uitvoeren van de meting' als het interpreteren of 'vergelijken met criteria'. De onderstaande tabel geeft de afbeelding van de Omega- op de ISO 14598-activiteiten.

Tabel 4.2 Vergelijking tussen de binnen de Omega-beoordelingsmethode uitgevoerde activiteiten en de door ISO 14598 onderkende activiteiten.

ISO 14598	Omega beoordeling
Bepaal doel van beoordeling	
Identificeer producttypen	
Specificeer kwaliteitsmodel	
Selecteer metrieken	Selecteren metrieken
Bepalen van kwalificatieniveaus voor de metrieken	Vaststellen normwaarden
Bepalen criteria voor de beoordeling	Vaststellen criteria
Formuleren van beoordelingsplan	
Uitvoeren van de metingen	Uitvoering
Vergelijken met criteria	Uitvoering
Vaststellen van de resultaten	

Deze afbeelding van activiteiten komt overeen met de aanpak van andere codemetriek gebaseerde benaderingen: er is aandacht voor metriekselectie, bepalen normwaarden, uitvoering en interpretatie van de metingen. Er wordt geen aandacht gegeven aan doelformulering en welke kwaliteit er beoordeeld moet worden.

Als we naar de overige subaspecten van aansturing kijken dan constateren we ten aanzien van de middelenallocatie dat deze toewijzing is uitgevoerd. Zo is aangegeven met welke middelen de beoordeling moet worden uitgevoerd. Er is bijvoorbeeld vastgelegd welke personen de meting uitvoeren en met welke tool dat gebeurt. De voortgangsbewaking ontbreekt echter. Er is niet gedefinieerd hoeveel tijd er per activiteit mag worden besteed en met welke inzet dat gebeurt. Ook is er geen controle of terugkoppelingsmechanisme voor onderkend.

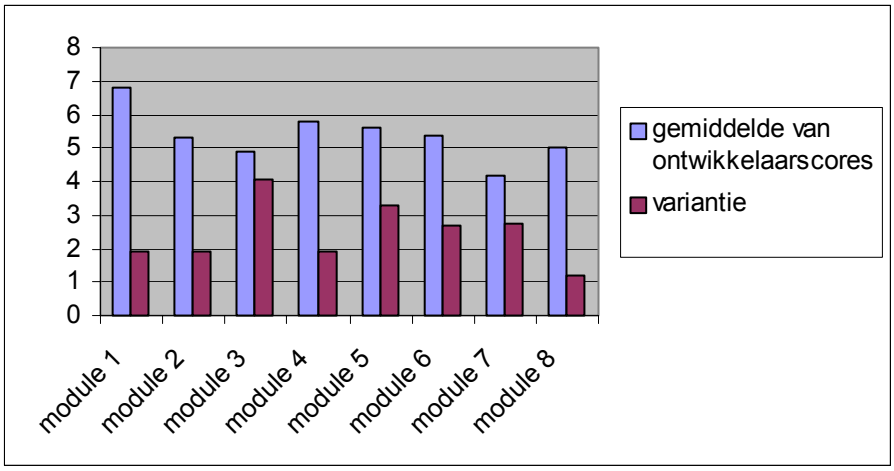
Interpretatie

De hiervoor gepresenteerde afbeelding van de beoordelingsactiviteiten op de ISO 14598 structuur laat zien dat er tijdens de Omega-beoordelingsmethode geen aandacht werd besteed aan vijf ISO activiteiten. Twee daarvan achten we wel van belang voor de beoordeling, namelijk: ‘bepaal doel van beoordeling’ en ‘specificeer kwaliteitsmodel’. Wat betreft de activiteit ‘bepaal doel van beoordeling’ is bij ervaringen rondom doelformulering al opgemerkt dat deze activiteit meer aandacht verdiend.

Wat betreft de uitvoering van de activiteit ‘specificeer kwaliteitsmodel’ komt het gebrek aan aansturing naar voren. Dat een Omega-module moet voldoen aan Maintainability is één stap. Maar wat dit inhoudt, ofwel wat Maintainability in de context van het Omega-systeem betekent, werd niet gedefinieerd. Maintainability bleef daarmee een abstract begrip. Binnen RPS is wel onderkend dat de karakteristiek geoperationaliseerd moest worden. Dit is echter niet van de grond gekomen omdat de ontwikkelaars onderling te sterk van mening

verschilden over wat onderhoudbare Omega-code was. In dit verband presenteren we resultaten die zijn opgedaan tijdens het experiment, dat in paragraaf 4.4 geïntroduceerd is.

Tijdens de uitvoering van het experiment viel op dat de ontwikkelaars onderling sterk van mening verschilden over de onderhoudbaarheid van een aantal modules. Deze verschillen van mening, of variantie, worden getoond in de onderstaande figuur. De figuur toont de gemiddelde ontwikkelaarscore nog een keer, maar nu ook met de variantie in de ontwikkelaarscores. De variantie is bepaald als de standaarddeviatie over de reeks scores.

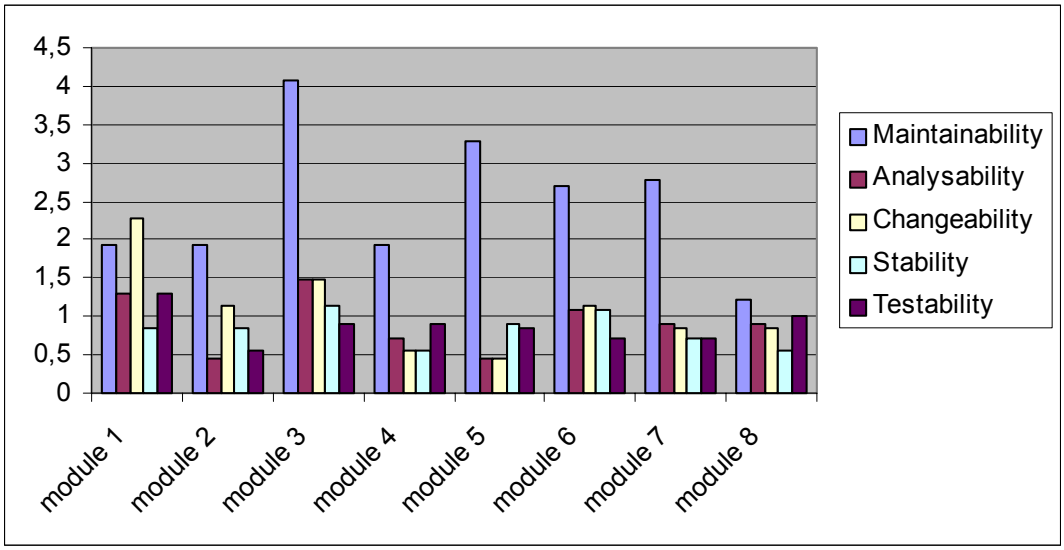


Figuur 4.4 Gemiddelde van de ontwikkelaarscores en de variantie daarin per module

De figuur laat zien dat het verschil tussen de meningen van de ontwikkelaars voor de modules erg groot is. Zo is er vrijwel steeds sprake van een variantie van rond de twee, maar voor module 3 is de variantie zelfs 4. Dit betekent dat bij het bepalen van de kwaliteit van deze module één ontwikkelaar is geweest die de module gescoord heeft als 8,9 (4,8 + 4) terwijl een andere persoon aan dezelfde module een waarde 0,9 heeft toegekend. Deze grote verschillen tussen de opinies van ontwikkelaars valt des te meer op als we zien dat de gemiddelde scores van de ontwikkelaars steeds tussen de 4 en 7 liggen. Op een schaal van 1 tot 10 is deze spreiding niet erg groot. Er zijn zo immers geen modules die bijzonder goed (9 of 10) of bijzonder slecht (1 of 2) scoren. Als we de ontwikkelaars apart nemen is dit wel het geval. We constateren dat er een grote variantie is in de meningen van de ontwikkelaars over dezelfde modules. Dit was te verwachten bij een dergelijke manier van scoren en dit is veelal ook de reden waarom de vraaggebaseerde beoordelingsbenadering als minder reproduceerbaar wordt beschouwd; zie hoofdstuk 2.

In dit verband valt op dat naarmate de ontwikkelaars naar hun mening over subkarakteristieken werd gevraagd de verschillen van mening verminderen. Tijdens het experiment is de ontwikkelaars namelijk gevraagd om per module aan te geven wat hun oordeel over de subkarakteristieken van onderhoudbaarheid (Maintainability) was. Het betrof de subkarakteristieken: Analysability, Changeability, Stability en Testability. Ook voor de

gemiddelde ontwikkelaarscores op deze subkarakteristieken is de variantie bepaald. Het resultaat is in de volgende figuur uitgebeeld.



Figuur 4.5 Variantie in gemiddelde ontwikkelaarscores voor Maintainability en vier subkarakteristieken

Het voorgaande leidt tot de constatering dat ontwikkelaars onderling minder van mening verschillen over de waarde van de subkarakteristieken dan over de hoofdkarakteristiek. Dit wijst erop dat de mate van detail waarmee ontwikkelaars naar hun mening wordt gevraagd van invloed is op de variantie in de beantwoording. Door ontwikkelaars in meer detail naar hun mening te vragen beperken we de variantie in de beantwoording. Dit sluit aan op de bevindingen van Xenos en Christostodoulakis (1997) in hun onderzoek naar het bepalen van kwaliteitskarakteristieken door uit te gaan van meningen.

4.7 Ervaringen met het afwegen van doel en middelen

De afwegingsproblematiek heeft betrekking op zowel doel als middelen. Op de doelformulering is al in paragraaf 4.5 ingegaan. In deze paragraaf beperken we ons daarom tot de middelen. Bij middelen ging het in de Omega-beoordelingsmethode om codemetrieken, normwaarden en het tool om de metingen uit te voeren.

De metrieken en normwaarden werden geselecteerd op basis van de overweging dat ze Maintainability meten. Dit betreft de ‘kwaliteit’ van de afweging; zie ook het analysekader in hoofdstuk 3. Bij deze afweging wordt ervan uitgegaan dat het feit dat deze metrieken en normwaarden worden voorgeschreven door het Logiscope referentiemodel voldoende reden is om ze ook in de Omega-beoordelingsmethode op te nemen. Van een kritische afweging is dan ook geen sprake.

Een ander element van afweging is ‘kwantiteit’. Dit speelde bij de Omega-afweging echter geen rol. Het bleef onbekend hoeveel tijd, geld en menskracht gemoeid zou zijn met de beoordeling. Hierbij moet worden bedacht dat het in deze beoordeling vooral gaat om het implementeren van het tool. Per metriek kostte dit gemiddeld drie dagen. Het toepassen van de metriek is geen middelenprobleem. Dit gebeurt immers geautomatiseerd. Een eventuele afweging rondom kwantiteit had zich dan ook moeten richten op de ontwikkeling van de beoordeling (activiteiten: selecteren metrieken, bepalen normwaarden en bepalen criteria).

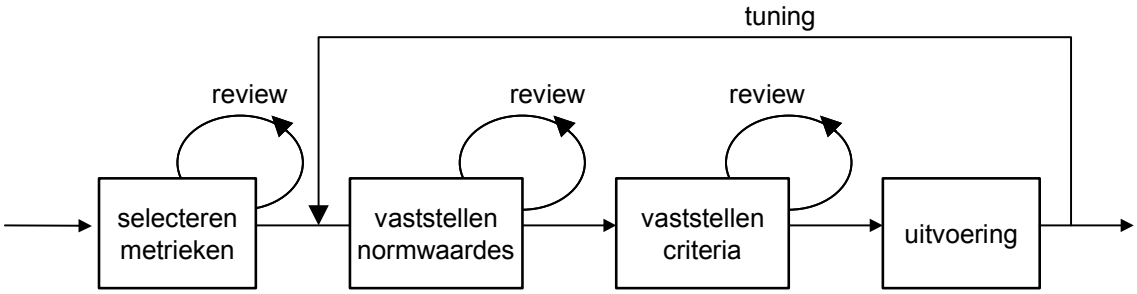
Wat betreft het tool wordt opgemerkt dat deze werd gekozen na vergelijking met andere tools. Het uiteindelijke tool werd gekozen vanwege de functionaliteit (er kan zowel code mee worden gecheckt als dat er mee gemeten kan worden) en prijs (het tool is relatief goedkoop). Bij deze afweging werd dus naar zowel kwantiteit als kwaliteit gekeken.

Interpretatie
We constateren dat de afweging van doel en middelen beperkt is. Er is wel enige aandacht besteed aan de ‘kwaliteit’, maar niet aan de ‘kwantiteit’ van de afweging. Zo worden metrieken geselecteerd omdat Maintainability gemeten moet worden (kwaliteit), maar de afweging op basis van tijd en kosten rondom de inzet van deze middelen (kwantiteit) blijft buiten beschouwing.

Ten aanzien van de selectie van metrieken en normwaarden valt op dat het vertrouwen in het Logiscope-tool doorslaggevend is. Het tool wordt gerespecteerd en de organisatie heeft vertrouwen in de metrieken die Logiscope voor het meten van Maintainability voorschrijft. Omdat hiermee wordt vertrouwd op de juistheid van het Logiscope referentiemodel in plaats van het zelf operationaliseren van wat Maintainability in de context van Omega betekent, vatten we dit op als beperkt rationeel.

4.8 Ervaringen met terugkoppeling en bijstellen van het proces

In de Omega-beoordelingsmethode vond de terugkoppeling op twee manieren plaats. Enerzijds ging het om de reviews die op de resultaten van de verschillende activiteiten werden uitgevoerd. Anderzijds ging het om het bijstellen van de normwaarden op basis van de gevonden meetresultaten. Dit gebeurde na de activiteit ‘uitvoeren beoordeling’. Dit laatste werd in de Omega-beoordelingsmethode aangeduid als tuning: het bijstellen van de normwaarden. Beide typen terugkoppeling werden binnen RPS niet zozeer als terugkoppeling gezien. Zo werd ‘tuning’ in eerste instantie als activiteit omschreven. De volgende figuur geeft een overzicht van de terugkoppeling in het beoordelingsproces.



Figuur 4.6 Terugkoppeling in de Omega-beoordelingsmethode

Review – dit betrof het overleg tussen de mensen die bij de ontwikkeling van de beoordeling zijn betrokken. Hierbij werden de resultaten van de activiteiten besproken, zoals de selectie van metrieken, de normwaarden en criteria. De tevredenheid en het vertrouwen over de resultaten was bepalend of er iets veranderde. Er werd niet expliciet vergeleken met het doel van de beoordeling. Er was daarmee niet echt sprake van terugkoppeling zoals gedefinieerd in het analysekader in hoofdstuk 3.

Tuning – de tuning van normwaarden was nodig omdat de Omega-beoordelingsmethode onderkende dat de uit het Logiscope referentiemodel overgenomen normwaarden nog moesten worden afgestemd op het product in kwestie. De tuning hield in dat normwaarden werden gewijzigd, waardoor ook het aantal functies dat volgens deze normwaarden 'onderhoudbaar' is, veranderde. De aantallen functies werden uitgedrukt in de categorieën 'uitstekend', 'goed', 'acceptabel' en 'slecht', zie figuur 4.1. Als de normwaarden werden gewijzigd, dan veranderden ook de percentages functies die tot de verschillende categorieën behoren. Deze percentages werden door de ontwikkelaars zelf geïnterpreteerd: zij bepaalden wat de optimale verdeling van functies over de categorieën was.

Interpretatie

We constateren dat er tijdens de Omega-beoordelingsmethode in beperkte mate wordt teruggekoppeld. De meest uitgewerkte terugkoppeling betreft het bijstellen van normwaarden, aangeduid als 'tuning'. Andere acties die als terugkoppeling worden opgevat zijn reviews. Maar deze zijn veel minder uitgewerkt.

Ten aanzien van de tuning constateren we dat er geen kennis over deze tuning werd opgebouwd. De argumentatie achter het kiezen van de waarden bleef daarmee grotendeels 'in de hoofden van de ontwikkelaars'. Zo kon tijdens de casestudie voor maar 4 van de 15 metrieken de motivering van de normwaarden worden achterhaald. Dit gold voor bijvoorbeeld de metriek LinesOfCode. Deze heeft als maximum normwaarde 60. Dit getal was terug te voeren op het aantal regels dat op een computerscherm past. Door functies ten hoogste 60 regels te laten zijn, past de gehele functie in één keer op het scherm. Dit is prettig voor de het analyseren van code. Dit werkt dus positief op de Analyseerbaarheid van de module. Voor de meeste andere metrieken was een dergelijke redenering niet te achterhalen.

We concluderen dan ook dat er tijdens de Omega-beoordelingsmethode weinig over het product en de daarop toe te passen metrieken en normwaarden is geleerd. Zo zijn de normwaarden bijgesteld gedurende de periode dat de beoordeling is uitgevoerd (ongeveer 1 jaar). Uit vraaggesprekken met ontwikkelaars bleek dat deze mensen niet het idee hadden dat er in die tijd betere normwaarden zijn opgesteld. Na 1 jaar voortdurend bijstellen van normwaarden bleken die waarden nog steeds niet naar tevredenheid iets over Maintainability te zeggen.

Verder concluderen we dat de beschreven terugkoppeling een beperkte scope heeft. De tuning betrof maar een deel van de beoordelingservaringen. Het ging alleen om de normwaarden, de keuze van de metrieken bleef buiten beschouwing. Er werd van uitgegaan dat de metrieken correct zijn. Bij ‘aansturing’ is echter opgemerkt dat Maintainability niet geoperationaliseerd is voor het Omega-systeem. Het is dan ook niet erg waarschijnlijk dat de set metrieken in één keer goed geselecteerd is. Vandaar dat het goed was geweest om ook terug te koppelen vanuit de ervaringen met de metrieken, om eventueel andere metrieken te selecteren.

4.9 Samenvatting

De Omega-casestudie laat zien dat de in hoofdstuk 3 geconstateerde problemen ten aanzien van de besturing van een beoordelingsproces relevant zijn.

Zo is geconstateerd dat de doelformulering onvoldoende is. Door de weinig expliciete formulering van het oorspronkelijke doel, zijn andere doelen ontstaan. Deze doelen worden echter niet allemaal gedekt door de uitwerking van de beoordeling. Hierdoor kan er ontevredenheid over de beoordeling ontstaan: men herkent in de beoordeling niet het na te streven doel.

Verder is er een gebrekkige aansturing van het proces geconstateerd. Een deel van de uitgevoerde activiteiten werd namelijk niet expliciet onderkend in de processtructuur. Er werd feitelijk maar één activiteit (uitvoering) echt onderkend, terwijl achteraf is vastgesteld dat er sprake was van vier activiteiten. Dit gebrek aan inzicht in het proces heeft gevolgen voor de inrichting. Het heeft wellicht tot gevolg dat bepaalde activiteiten werden overgeslagen. Zo heeft er geen operationalisering van het begrip Maintainability plaatsgevonden. Het gebrek aan processtructuur maakt voortgangsbewaking onmogelijk, terwijl de middelenallocatie zich beperkt tot de expliciet onderkende activiteit.

Wat betreft de afweging tussen doel en middelen is geconstateerd dat deze beperkt is geweest. Er is wel enigszins aandacht besteed aan de ‘kwaliteit’, maar niet aan de ‘kwantiteit’ van de afweging. Zo worden metrieken geselecteerd omdat Maintainability

gemeten moet worden (kwaliteit), maar blijven de tijd en kosten van deze middelen (kwantiteit) buiten beschouwing.

Tenslotte is vastgesteld dat er tijdens de beoordeling onvoldoende wordt teruggekoppeld. De meest uitgewerkte terugkoppeling betreft het bijstellen van normwaarden, aangeduid als ‘tuning’. Andere acties die als terugkoppeling worden opgevat zijn reviews. Maar dit is veel minder uitgewerkt. Omdat de ‘tuning’ alleen de normwaarden betreft, concluderen we dat de scope van de terugkoppeling beperkt is.

5. HIS casestudie

Het doel van deze casestudie is om aan te geven dat de vier problemen die zijn geïdentificeerd in hoofdstuk 3, relevant zijn in de beoordelingspraktijk. De casestudie betreft de ontwikkeling van een beoordelingsaanpak voor Huisartsinformatiesystemen (HIS-en).

5.1 Verantwoording

Huisartsinformatiesystemen (HIS) en de achtergronden van de beoordeling ervan worden beschreven in paragraaf 5.2. De beoordelingsaanpak zelf komt in paragraaf 5.3 aan de orde. Vervolgens worden de ervaringen met de beoordelingsaanpak behandeld.

Rol van de onderzoeker

Uitgangspunt van de casestudie was kritiek op de HIS-beoordelingsmethode. Het opmerkelijke aan deze kritiek was dat ze geventileerd werd tijdens de ontwikkeling van de methode. Dit fascineerde de onderzoeker, omdat er kritiek op de beoordeling bestond zonder dat er ook maar één keer een beoordeling was uitgevoerd.

De beoordeling werd in opdracht van het Nederlands Huisartsen Genootschap (NHG), ontwikkeld door KEMA Nederland. De onderzoeker had als medewerker van KEMA toegang tot het team dat de beoordeling ontwikkelde en kon zo alles van nabij volgen. Hij was hierbij niet verantwoordelijk voor ontwikkelingstaken, maar heeft wel meegewerkt door advies te geven over het ontwikkelen van checklists die tijdens de beoordeling worden gebruikt.

Omdat de onderzoeker niet verantwoordelijk was voor de ontwikkeling van de beoordeling en via zijn functie toch toegang had tot informatiebronnen kon de casestudie direct worden uitgevoerd. In tegenstelling tot de Omega casestudie, die behandeld is in hoofdstuk 4, was het dan ook niet nodig om vooraf aan de casestudie een adviesopdracht uit te voeren. De onderzoeker kon zich dan ook als onderzoeker opstellen en het analysekader direct toepassen.

Aanpak en informatiebronnen

De casus wordt eerst kort beschreven. Vervolgens komt een analyse van de kritiek op het beoordelingsproces. Het was immers deze kritiek die de aanleiding voor de casestudie vormde. Om de kritiek te analyseren zijn de bij de beoordeling betrokken partijen geïdentificeerd en is bepaald welke kritiek deze partijen op de beoordeling hadden. Deze analyse van de HIS-beoordelingsmethode is een opmaat voor de analyse van de vier

problemen rondom de besturing van de beoordeling. Deze tweede analyse vormde het doel van de casestudie. Hiervoor werd gekeken naar de ervaringen tijdens de HIS-beoordelingsmethode met:

- Doelformulering – hoe wordt doel opgesteld?
- Aansturing van proces – welke activiteiten worden beschreven en hoe hangen ze samen?
- Afwegen van doel en middelen – hoe worden de middelen geselecteerd, welke afwegingen worden daarbij gemaakt?
- Terugkoppeling en bijsturing – hoe wordt (eventuele) terugkoppeling uitgevoerd?

Deze aspecten zijn ingevuld op basis van observatie van de onderzoeker. Tijdens de uitvoering van de casestudie is de beoordeling één keer toegepast. De ervaringen opgedaan tijdens de casestudie zijn dan ook voornamelijk gebaseerd op de gang van zaken rondom de ontwikkeling van de beoordeling. In termen van de ISO 14598 processtructuur betekent dit dat de activiteiten 'bepalen doel van beoordeling' tot en met 'bepalen criteria' aan de orde komen.

Tijdens de casestudie zijn vier informatiebronnen geraadpleegd, namelijk:

- specialisten van KEMA (de ontwikkelaars van de beoordeling),
- interne notities van het overleg tussen leveranciers en KEMA enerzijds en NHG – KEMA anderzijds,
- beoordelingsdocumentatie; documenten waarin het beoordelingsproces wordt beschreven en
- de berichtgeving over huisartsinformatiesystemen in de Automatisering Gids. Het betrof hier berichten vanaf 1998 tot op heden waarin zowel over de ontwikkeling als de beoordeling van Huisartsinformatiesystemen werd verhaald.

De resultaten van deze analyse worden in de paragrafen 5.5 tot en met 5.8 gepresenteerd. Eerst wordt ingegaan op Huisarts informatiesystemen (paragraaf 5.2), de opzet van de beoordeling (paragraaf 5.3) en de kritiek op de beoordeling (paragraaf 5.4).

5.2 Huisarts-informatiesysteem (HIS)

Sinds 1980 zijn huisartsen in Nederland bezig geweest met het automatiseren van hun praktijk. Huisartsen en -praktijken gebruiken hiervoor Huisartsinformatiesystemen. Dit wordt veelal afgekort tot HIS-en. In het begin was de automatisering beperkt tot een groep ‘early adaptors’, die alleen hun (patiënt)administratie automatiseerden met behulp van de computer. Anno 2000 gebruiken –vrijwel– alle huisartsen in Nederland een HIS. In de loop der jaren is de functionaliteit van de HIS-en gegroeid en tegenwoordig wordt een groot deel van de handelingen van een huisarts ondersteund door een dergelijk informatiesysteem. Een HIS omvat tegenwoordig onder andere de volgende functionaliteit: het uitschrijven van recepten, apotheek beheer, elektronisch declareren naar ziektekostenverzekeraars.

In de loop der jaren zijn er vanuit de huisartsenpraktijken eisen aan HIS-en gesteld. Hiervoor is in 1985 de Werkgroep Coördinatie Informatisering Automatisering (WCIA) opgericht. Initiatiefnemers waren het Nederlands Huisartsen Genootschap (NHG) en de Landelijke Huisartsen Vereniging (LHV). De NHG is de branchevereniging voor huisartsen die zich richt op het up-to-date houden van de vakinhoudelijke kennis van huisartsen. De vereniging heeft tot doel om de Nederlandse huisartsen op een bepaald kwaliteitsniveau te laten functioneren. De LHV is de vakbond van huisartsen en meer de belangenvereniging van de huisartsen. Beide zijn vertegenwoordigers van de huisartsen. De WCIA werd door zowel LHV als NHG opgevat als een dienstverlening naar de huisartspraktijken om antwoord te geven op de vraag 'wat goede en slechte HIS-en zijn'. Via de WCIA stellen huisartsen en hun verenigingen eisen aan huisartsinformatiesystemen.

Op basis van de eisen gesteld in de WCIA zijn in het verleden verschillende huisartsinformatiesystemen getoetst. Deze toetsing is geïnitieerd door de NHG. De LHV was hier niet bij betrokken. In het midden van de jaren '90 kregen huisartsen een automatiseringsvergoeding van de zorgverzekeraars. Deze vergoeding gold alleen in het geval van gebruik van HIS-en die aan WCIA-eisen voldeden. De huisartsen vroegen daarom aan de leveranciers om hun systemen te ontwikkelen én te toetsen op basis van de WCIA-eisen. Deze toetsing werd uitgevoerd door de NHG. Eind jaren '90 werd het voldoen aan de WCIA-eisen afgeschaft als voorwaarde voor het in aanmerking komen voor de automatiseringsvergoeding. De WCIA-eisen waren toen een 'de facto' standaard geworden: leveranciers boden hun systemen inmiddels vrijwillig aan ter toetsing.

Sinds 1985 zijn er drie versies van het WCIA-referentiemodel opgesteld. De laatste versie betreft het WCIA-HIS referentiemodel-1995 (WCIA, 1996). Dit referentiemodel omvat 800 –voornamelijk functionele– eisen aan een huisarts-informatiesysteem. Door de omvang heeft de ontwikkeling ervan vertraging opgelopen. Het uiteindelijke model is pas in 1997 gepubliceerd. Om de toetsing van HIS-en aan de eisen uit dit model goed te laten verlopen heeft de NHG in 1997 KEMA Nederland aangezocht om een nieuwe aanpak voor de HIS-beoordelingsmethode te ontwikkelen (AG 28 mei 1998). Naast de omvang van het eisenpakket speelde ook dat de beoordelingen op basis van de oudere versies van het referentiemodel problemen opleverden. De doorlooptijd van de beoordelingen was te lang en de beoordeling werd als slecht onderbouwd en té kwalitatief ervaren. Verder is KEMA aangezocht omdat de organisatie ervaring heeft met de 'state-of-the-art' rondom softwareproduct beoordelen van softwareproducten (toepassing van ISO-standaard) en het certificeren van producten en diensten. De NHG streeft op termijn naar het keuren en het vervolgens afgeven van een keurmerk voor HIS-en.

5.3 Beoordeling van huisartsinformatiesystemen

De ontwikkeling van het HIS-beoordelingsmethode is gestart met het opstellen van een kwaliteitsprofiel. Met dit profiel werd bepaald in welke mate een huisarts-informatiesysteem aan welke kwaliteitskarakteristieken moet voldoen.

Bij het opstellen van dit kwaliteitsprofielen is gebruik gemaakt van de Space-Ufo-methode die door KEMA, TU Eindhoven en andere partners binnen het Europese Space-Ufoproject zijn ontwikkeld (Eisinga e.a., 1995), (Space-Ufo consortium, 1998), (Van Ekris, 1998). Hierbij wordt de belanghebbenden bij het product gevraagd naar wat zij belangrijk aan het product vinden. Deze interviews zijn gehouden aan de hand van een gestructureerde vragenlijst. De antwoorden werden vervolgens met behulp van een matrix omgewerkt tot een prioritering van de kwaliteitskarakteristieken op de niveaus A, B, C of D. De gebruikte matrix is gebaseerd op inzichten van experts in wat belangrijke factoren bij de verschillende kwaliteitskarakteristieken zijn. Voor het opstellen van het kwaliteitsprofiel voor HIS-en werd een aantal artsen en artsassistenten geïnterviewd. Dit leverde uiteindelijk het kwaliteitsprofiel op dat in de volgende figuur wordt afgebeeld.

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

Figuur 5.1 Kwaliteitsprofiel van een huisarts-informatiesysteem

Het bovenstaande kwaliteitsprofiel laat zien dat de kwaliteitskarakteristiek Functionality erg belangrijk wordt gevonden voor een HIS. Dit komt vooral omdat de aanwezigheid en correctheid van de aanwezige –en in het WCIA vereiste– functies moet worden bepaald. Daarnaast zijn de karakteristieken Usability, Reliability en Efficiency van belang. Het kwaliteitsprofiel maakt duidelijk dat de artsen en artsassistenten de karakteristieken Maintainability en Portability minder belangrijk vinden.

Om tot oordelen over de verschillende kwaliteitskarakteristieken te komen werd een aantal checklists ontwikkeld. Deze checklists bestonden uit een aantal vragen met elk 4 antwoordmogelijkheden. Deze combinaties van vraag en scoremogelijkheden werden aangeduid als checklist items.

Voorbeelden van ontwikkelde checklists zijn de checklist ‘Aanwezigheid functies’ en de checklist ‘Documentatie’. Met de eerste checklist werd nagegaan of de vereiste functies geïmplementeerd zijn. Voorbeelden van vragen waren: ‘is er een diagnostische functie?’, ‘is

er een farmacie functie?'. Ook de checklist 'Documentatie' ging in op de karakteristiek Functionality, maar richtte zich op Suitability, een subkarakteristiek van Functionality. Bij de interpretatie van deze karakteristiek werd er vanuit gegaan dat het voor de gebruiker duidelijk diende te zijn welke functionaliteit door het HIS werd geboden om te kunnen bepalen of het pakket geschikt (suitable) is. Hiervoor moet de product- en gebruikersdocumentatie eenduidig en consistent zijn en een goede mate van detail hebben. Deze interpretatie resulteerde in vragen als 'is er een deugdelijke productbeschrijving die de mogelijkheden van het pakket beschijft?' en 'zijn alle beperkende voorwaarden van het systeem hierin vastgelegd?'

De opgestelde checklists dienden om een oordeel per kwaliteits(sub)karakteristiek van een specifiek HIS te kunnen geven. Dit gebeurde door het beantwoorden van de afzonderlijke vragen. Voor elke vraag werden namelijk vier scoremogelijkheden opgesteld: niet relevant (NR), niet voldaan (NS), deels voldaan (PS), volledig voldaan (FS). Aan elke mogelijkheid hing een waarde, variërend van 0 (niet voldaan), 1 (deels voldaan) tot 2 (volledig voldaan). Per item werd een gewicht vastgesteld. Dit gewicht bepaalde het relatieve belang van ieder item. Door alle vragen van een checklist te beantwoorden werden alle waarden bepaald. Door het toepassen van de gewichten (per vraag) kon uiteindelijk de eindwaarde van de kwaliteitskarakteristiek worden bepaald.

5.4 Kritiek op de beoordeling

Al tijdens de ontwikkeling werd er kritiek op de HIS-beoordelingsmethode geuit. Deze kritiek was onder meer afkomstig van de leveranciers en de HIS-gebruikersvereniging. Beide vonden 'veel eisen te zwaar'. Ook zouden veel eisen 'te weinig eenduidig zijn gesteld'. Daarnaast vonden de leveranciers de toetsing te duur. Dit commentaar werd publiekelijk geuit in het vakblad 'Automatisering gids'. Vanaf 1998 verschenen verschillende artikelen met dergelijke alarmerende berichten over huisartsinformatiesystemen, de WCIA-eisen en de HIS-beoordelingsmethode (AG 15 januari 1999; AG 5 februari 1999; AG 28 januari 2000). De artikelen uit de 'Automatisering gids' zijn als uitgangspunt genomen voor de analyse van de kritiek. De kritiek is afkomstig van verschillende partijen. Om hier meer helderheid in te verschaffen worden de verschillende partijen eerst geanalyseerd. Naast de al onderkende NHG, LHV en KEMA zijn er de leveranciers en HIS-gebruikers en nog andere partijen. Deze partijen worden hieronder in kaart gebracht.

Partijen

Een sterk betrokken partij betreft de leveranciers van de HIS-pakketten. Dit zijn de makers van pakketten als Topaas, MacroHis en HetHis. Deze leveranciers worden geacht om hun pakketten ter keuring aan te bieden. Hiermee ontstond er een relatie tussen leveranciers, KEMA en NHG. De laatste partij onderhoudt van oudsher de relatie met de leveranciers en stelt ook de eisen aan het te ontwikkelen product. KEMA diende de beoordeling van het

pakket van de leverancier uit te voeren op basis van de door NHG –via WCIA– gestelde eisen.

Nedhis is de overkoepelende organisatie van de gebruikersverenigingen van de diverse HIS-pakketten. Het betreft, net als NHG en LHV, een vertegenwoordiger van de huisartsen. Nedhis redeneert vanuit het perspectief 'automatisering voor de huisartsen'.

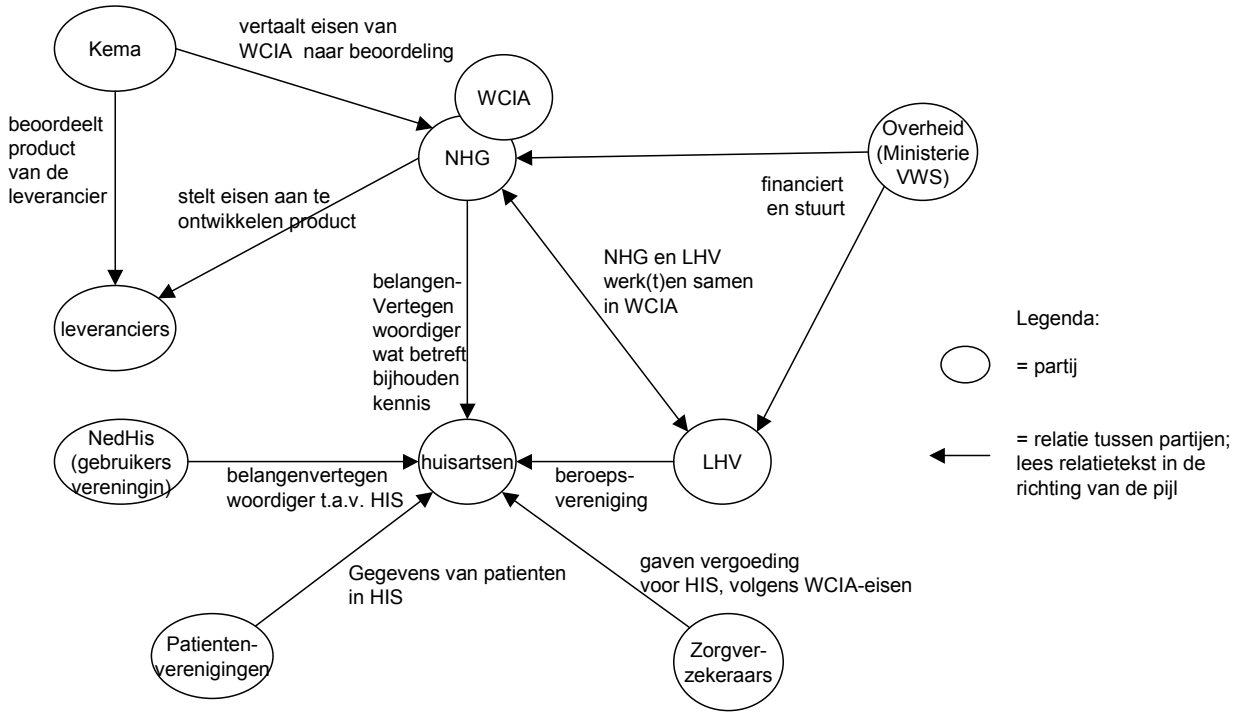
Er zijn zo drie vertegenwoordigers van de huisartsen rondom de HIS-beoordelingsmethode onderkend. Elke vertegenwoordiger komt vanuit een bepaald perspectief op voor de belangen van huisartsen. De Nedhis dus voor de automatisering van de huisartspraktijk, de NHG voor het up-to-date houden van de kennis van huisartsen en de LHV voor algemene en financiële belangen van huisartsen. Het is opvallend dat de HIS-beoordelingsmethode niet is geïnitieerd of wordt gedragen door de Nedhis. Dit zou qua belangenvertegenwoordiger de meest logische partij zijn geweest. In plaats daarvan is de NHG de drijvende kracht achter de beoordeling. Dit is te verklaren uit het ontstaan van WCIA: het pakket eisen waarop HIS-en beoordeeld worden. Deze eisen zijn niet door de Nedhis geformuleerd, maar door NHG (en LHV). Uit het eisenpakket is de beoordelingsmethode ontstaan.

Ook de overheid is een partij in de HIS-beoordelingsmethode. De Nederlandse overheid bemoeit zich op afstand met de ontwikkeling van de automatisering van huisartspraktijken. Het Ministerie van Volksgezondheid, Welzijn en Sport (VWS) spreekt met de LHV, als vertegenwoordiger van huisartsen, maar ook met de NHG als het gaat om het 'algemeen kwaliteits- of kennisniveau van de huisartsen'. De overheid stuurt door formele besluiten te nemen, zoals het vaststellen van behandeltarieven, vergoedingen van medicijnen, enzovoorts. Een voorbeeld is het schrappen van vergoedingen voor het eenvoudige geneesmiddelen die zonder recept zijn te verkrijgen. Alleen middelen voor chronisch gebruik worden vergoed. Bij chronisch gebruik moet de huisarts dit met de code CG op het recept aanbrengen. Het HIS moet hierop worden aangepast (AG 20 augustus 1999). Ook stimuleert ze beleid door projecten te subsidiëren, zoals het Elektronisch Patiëntendossier (AG 23 april 1999). De Nederlandse overheid heeft het opzetten en uitvoeren van HIS-beoordelingen niet gestimuleerd. Zo is er in Nederland geen wettelijke verplichting voor leveranciers om aan de (WCIA) eisen te voldoen. De leveranciers bieden hun pakket vrijwillig ter keuring aan. In bijvoorbeeld Ierland is er wel een wettelijke verplichting (McMullan, 1994).

De zorgverzekeraars zijn ook een partij in de HIS-beoordelingen. Door in de jaren '90 alleen de kosten van volgens de WCIA goedgekeurde HIS-pakketten te vergoeden hebben de verzekeraars ertoe bijgedragen dat de HIS-beoordelingsmethode op basis van de WCIA een standaard werd. De verzekeraars hebben zich hierbij overigens niet actief met de beoordeling bemoeid.

Tenslotte noemen we de patiëntenverenigingen als partij. De patiënten zijn immers de mensen wier data in de HIS-en worden opgeslagen. Uit de bestudeerde documentatie rondom het WCIA en de HIS-beoordelingsmethode (WCIA, 1996) en de uitgevoerde interviews is niet naar voren gekomen dat de patiëntenverenigingen zich actief bemoeid hebben met de HIS-beoordelingsmethode.

De onderstaande figuur geeft de hiervoor beschreven partijen weer en de relaties daartussen.



Figuur 5.2 Partijen betrokken bij de HIS-ontwikkeling en beoordeling

Kritiek op de beoordeling

Van de genoemde partijen hebben de leveranciers, de NHG, KEMA en Nedhis kritiek op de beoordeling geuit. De leveranciers en de Nedhis zijn het meest kritisch geweest, maar ook KEMA en NHG hebben tijdens de ontwikkeling kritiek geuit. Onze analyse van de uitlatingen en de gang van zaken tijdens de beoordeling leidt tot drie punten van kritiek:

- eisen zijn te zwaar,
- twijfel over validiteit van checklists,
- verhouding prijs en prestatie van de beoordeling niet goed.

Eisen zijn te zwaar – de gebruikersvereniging (Nedhis) en de leveranciers hadden commentaar op de WCIA-eisen die aan HIS-en werden gesteld en die terugkwamen in de beoordeling. Leveranciers vonden veel eisen ‘te zwaar’ en vonden daarnaast veel eisen ‘te weinig eenduidig gesteld’ (AG 5 februari 1999; AG 18 februari 2000)

De Nedhis vond eveneens dat de meetlat te hoog werd gelegd. Technische en functionele eisen waren te zwaar opgeschroefd (AG 28 januari 2000).

De kritiek dat de eisen te zwaar zijn is opvallend omdat de HIS-beoordelingsmethode juist werd opgesteld vanuit het besef dat de WCIA-eisen te zwaar en te kwalitatief (lees: te moeilijk en niet eenduidig) waren. De kritiek moet dan ook deels worden geplaatst in het licht van het politieke krachtenveld waarin de verschillende partijen opereerden. In dit krachtenveld stonden zowel Nedhis als de leveranciers tegenover de NHG. De NHG had de WCIA mede opgesteld, de leveranciers en Nedhis zetten zich hier tegen af. Hierbij valt op dat de NHG de vertegenwoordiger van de WCIA was, de LHV was er kennelijk niet –meer– bij betrokken. Dit heeft te maken met het feit dat het NHG de 'motor' achter de HIS-beoordelingsmethode was. De LHV heeft wel geparticipeerd bij het opstellen van de WCIA-eisen, maar stond verder buiten de HIS-beoordeling.

Twijfel over de validiteit van de checklists – tijdens het ontwikkelen van de beoordeling zijn er bij de KEMA en de NHG twijfels ontstaan over de voor de beoordeling benodigde checklists. Het betrof allereerst de validiteit van de checklists. Het ging hierbij om de vraag of 'er met de checklists werd gemeten wat men wilde meten?'. Na een zorgvuldige ontwikkeling van de checklists, waarbij gebruik werd gemaakt van 'state-of-the-art' kennis bestonden er nog steeds twijfels. Zelfs tijdens het reviewen van de checklists ervoeren KEMA- en NHG-reviewers problemen met het verband tussen de items van de checklist en kwaliteitskarakteristieken. De problemen betroffen: volledigheid (worden alle relevante items gesteld), waardering (hoe een gewicht toekennen aan items) en bepalen normwaarden (wanneer werkt een antwoord op een vraag positief door op de karakteristiek en wanneer niet).

Deze twijfels over de checklists waren nota bene afkomstig van de beoordelende instantie, KEMA zelf. Dit was dan ook niet zozeer een gevolg van onkunde, als wel van het structurele probleem rondom het meten van externe kwaliteit (de kwaliteitskarakteristieken) met interne metrieken (de checklist vragen). Dit probleem is al in hoofdstuk 2 aangeduid. Het is vrijwel onmogelijk om tot een eenduidige interpretatie van kwaliteitskarakteristieken te komen.

Verhouding prijs en prestatie van de beoordeling is niet goed – de leveranciers hadden kritiek op de prijs van de beoordeling: de toetsing is qua prijs verdrievoudigd t.o.v. de beoordeling met de voorgaande versie van het WCIA (AG 5 februari, 1999). Dit terwijl één van de doelen van de nieuwe beoordelingsmethode juist was om doorlooptijd en daarmee de kosten te reduceren. De prijsstijging kwam vooral door de toegenomen omvang van de WCIA-eisen. Ook al werd het totaal aantal eisen (800) gereduceerd tot een meer beperkte set in de checklists, er bleef nog altijd een groot aantal (rond de 500) vragen over. Het toepassen van deze checklists kostte daarmee veel tijd en inzet van arbeid. Dit vertaalde zich in de prijs van de beoordeling.

Deze kritiek op de prijs van de HIS-beoordelingen wijst in de richting van de balanceerproblematiek. Er zijn doorwrochte beoordelingstechnieken ontwikkeld, maar de partij die er voor zal moeten betalen heeft grote moeite met het betalen van de prijs ervoor.

Conclusie – voorgaande analyse maakt duidelijk dat er vanuit verschillende partijen kritiek werd geuit op de HIS-beoordelingsmethode. Een deel van deze kritiek is te verklaren uit belangentegenstellingen: HIS-leveranciers en Nedhis versus de NHG. Ondanks problemen met het opstellen van de checklists, is het niet zo dat er sprake is van een slechte beoordeling. Zo is er een geavanceerde methode voor het bepalen van het kwaliteitsprofiel gebruikt, is er een onderbouwd beoordelingsplan opgesteld en toegepast. We constateren dat de beoordeling voldoet aan wat er van een vraaggebaseerde beoordeling verwacht mag worden.

De kritiek maakt echter duidelijk dat er problemen zijn. Eén van de problemen (de validiteit van de checklists) heeft te maken met een structureel meetprobleem, namelijk het bepalen van externe kwaliteit met behulp van interne metrieken. De twee andere problemen zijn terug te voeren op de besturing van het beoordelingsproces. Hiermee komen we op de problemen rondom de besturing van het beoordelingsproces die in hoofdstuk 3 reeds zijn opgesteld. In de volgende paragrafen komen de ervaringen rondom elk van daar benoemde probleemgebieden aan de orde.

5.5 Ervaringen met het formuleren van doel van beoordeling

Het doel van de HIS-beoordelingsmethode wordt geformuleerd als ‘het verkrijgen van vertrouwen in de goede en juiste werking van het HIS om vast te stellen of de aanwezige functionaliteit voldoet aan de minimum eisen van het WCIA-HIS-referentiemodel’ (WCIA, 1999). De HIS-en dienen beoordeeld te worden op basis van de in het referentiemodel beschreven functionele eisen. Het beoordelen van HIS-pakketten en het vervolgens afgeven van een keurmerk aan het pakket wordt door het NHG van belang geacht vanwege het toenemende belang van software in de huisartsenwereld en de toenemende eisen die aan software worden gesteld. De beoordeling is bedoeld om een bepaald kwaliteitsniveau van het pakket te garanderen. Daarnaast speelde de verkorting van de doorlooptijd van de beoordeling een rol. De beoordeling werd immers (opnieuw) ontwikkeld omdat de voorgaande te uitgebreid was. Van deze twee doelstellingen werd alleen de eerste geconcretiseerd. Een deel van het beoordelingsdoel –namelijk verkorting van de doorlooptijd– is daarmee verdwenen!

Doelformulering middels kwaliteitsprofiel

Om deze abstracte doelstelling concreter te maken, werd voor de HIS-beoordelingsmethode een kwaliteitsprofiel opgesteld. Zo werd het doel in termen van kwaliteitskarakteristieken,

Functionality, Reliability, enzovoorts, en bijbehorende beoordelingsniveaus –A, B, C, D– uitgedrukt.

Met het kwaliteitsprofiel werd het doel van beoordeling geoperationaliseerd. Wel wordt opgemerkt dat de kwaliteitkarakteristieken in feite eigenschappen van het product betreffen. Dit is per definitie wat anders dan het doel van beoordeling. De gang van zaken tijdens de HIS-beoordelingsmethode laat zien dat het kennelijk nodig is om het doel scherper te definiëren door het te verwoorden als eigenschappen en beoordelingsniveaus, het kwaliteitsprofiel.

Het geformuleerde kwaliteitsprofiel staat in de kritiek op de HIS-beoordelingsmethode niet ter discussie. Het kwaliteitsprofiel wordt positief ontvangen. De verschillende belanghebbenden beschouwden het profiel als een goede weergave van de relevante eigenschappen.

Ten aanzien van het middels de Space-Ufo methode opgestelde kwaliteitsprofiel valt op dat er uiteindelijk eisen zijn toegevoegd. Zo moet het product ook voldoen aan zogenaamde intake eisen. Dit betreft de vraag of alle onderdelen voor de beoordeling aanwezig zijn. Verder moet er voldaan worden aan de Euro/millenium conversie-eisen en dient de HIS-leverancier te voldoen aan (ISO 9000-3) proceseisen. Deze eisen zijn door NHG en KEMA toegevoegd aan de eigenschappen die volgen uit het kwaliteitsprofiel.

Betrokkenheid van partijen bij doelformulering
In paragraaf 5.4 zijn de partijen geschetst die in meer of mindere mate bij de beoordelingen betrokken zijn. Van deze partijen heeft de NHG het doel geformuleerd. Dat alleen de NHG, en niet de leveranciers en de Nedhis betrokken waren, heeft gevolgen voor de doelformulering.

In de algemene doelformulering wordt aangegeven dat HIS-en beoordeeld worden op basis van functionele eisen die zijn beschreven in het WCIA-HIS-referentiemodel. Dat een HIS op deze eisen wordt beoordeeld is logisch gezien de positie van de NHG, die bij de ontwikkeling van de WCIA-eisen betrokken is geweest. Gezien de kritiek die Nedhis en de HIS-leveranciers op de WCIA hebben (zie hiervoor) valt te verwachten dat als deze partijen bij de formulering betrokken zouden zijn geweest er een andere formulering was gekozen.

Ook bij de uitwerking (operationaliseren) van het algemene doel in het kwaliteitsprofiel speelt de selectie van betrokken partijen een rol. Het kwaliteitsprofiel wordt namelijk opgesteld door huisartsen te raadplegen. De leveranciers staan hier buiten. Met het negeren van de leveranciers wordt een invalshoek weggelaten. Zeker omdat HIS een moeilijk te ontwikkelen systeem is zou de kennis van de leveranciers van groot belang kunnen zijn. Dat de ontwikkeling van HIS-en moeilijk en zelfs problematisch verloopt laat de ontwikkeling

van Topaas zien. De ontwikkeling van het pakket is begin 2000 stopgezet door SMS Cendata en overgedragen aan de huisartsen zelf (AG 3 maart 2000), (AG 23 juni 2000). De leverancier ziet geen brood meer in het pakket door het uitlopen van de ontwikkelingstijd van het pakket en levert dan de broncode en documentatie aan de gebruikers. Deze mogen er zelf zien uit te komen. Interessant in dit verband is nog de visie van een oud-medewerker van één van de HIS-leveranciers die de LHV en NHG aanduidt als 'kibbelende gebruikers' die de HIS-ontwikkeling verlammen (AG 14 juli 2000)

Naast het probleem van het ontbreken van de inbreng van de kennis van de leveranciers is een tweede probleem de hieruit resulterende beperking van het draagvlak voor acceptatie van de beoordeling. De leveranciers zullen voldoen aan de WCIA-eisen omdat NHG en huisartsen dit eisen, maar ze staan er niet echt achter. Dit vormt de basis van de in paragraaf 5.4 geciteerde kritiek dat 'eisen onnodig hoog zijn' en 'te weinig eenduidig zijn'.

5.6 Ervaringen met het aansturen van het proces

Het aspect aansturen van proces betreft drie subaspecten, namelijk: inrichting van proces, toewijzen van middelen en voortgangsbewaking.

We gaan hieronder eerst in op de inrichting van het proces en stellen vast dat de volgende activiteiten tijdens de ontwikkeling van de HIS-beoordelingsmethode aan de orde kwamen, namelijk:

1. bepalen van kwaliteitsprofiel
2. kiezen van technieken
3. construeren van checklists
4. uitvoeren van beoordeling

Bepalen van kwaliteitsprofiel – tijdens deze activiteit werd vastgesteld welke kwaliteitskarakteristieken en welke beoordelingsniveaus van belang zijn voor een HIS. In paragraaf 5.5 is aangegeven dat hiervoor de Space-Ufo-methode is gebruikt. Verder is aangegeven dat met het kwaliteitsprofiel het beoordelingsdoel wordt geoperationaliseerd. De activiteit heeft daarmee betrekking op de ISO 14598-activiteiten 'bepaal doel van beoordeling' en 'specificeer kwaliteitsmodel'.

Keuze van technieken – na het opstellen van het kwaliteitsprofiel is bepaald met welke technieken de beoordeling moet worden uitgevoerd. Voor een groot aantal (sub)karakteristieken is hierbij voor een checklist gekozen. Daarnaast wordt er voorgesteld om de eigenschappen Functionality en Usability ook met een dynamische techniek te bepalen. Een dergelijke techniek wordt toegepast op een werkend systeem. Voor de HIS-beoordelingsmethode is gekozen voor een checklist waarin de handelingen van de gebruikers zijn beschreven. Door deze 'dynamische' checklist te doorlopen wordt het gedrag van het

systeem in kaart gebracht. Er werd gekozen voor het toepassen van dynamische technieken voor Functionality en Usability, omdat deze karakteristieken als de meest belangrijke werden gezien. De activiteit ‘keuze van technieken’ komt nog het meest overeen met ISO 14598 activiteit ‘selecteer metrieken’.

Construeren van checklists – deze activiteit betreft het opstellen van de –statische alsook dynamische– checklists voor het beoordelen van de kwaliteitskarakteristieken die in het kwaliteitsprofiel zijn vastgesteld. Deze checklists bestaan uit een aantal vragen (of items) waaraan scoremogelijkheden zijn gekoppeld. In paragraaf 5.3 is hierop ook ingegaan. De items werden geselecteerd uit bestaande (KEMA) checklists, uit het WCIA-referentiemodel en uit de ISO 12119-standaard. Items werden uit deze referenties overgenomen voor zover ze door de opstellers van de checklists relevant geacht werden voor het beoordelen van HIS-en. De selectie is dus gebaseerd op het inzicht en expertise van deze betrokkenen.

Beoordeling uitvoeren – de beoordeling zal worden uitgevoerd door de checklists toe te passen op het te beoordelen HIS. De items van de checklists worden beantwoord door een (onafhankelijke)beoordelaar. Deze persoon bepaald per item een antwoord. Hierbij heeft de beoordelaar een bepaalde (interpretatie)vrijheid. Een voorbeeld is de vraag ‘is in de productdocumentatie beschreven wat de diverse mogelijkheden zijn ten aanzien van beveiliging?’. Bij de beantwoording van deze vraag bestaat de interpretatievrijheid uit de invulling van wat er onder ‘diverse mogelijkheden’ wordt verstaan. Als alle items van een checklist zijn beantwoord, wordt de eindscore berekend. De antwoorden op elk van de vragen worden volgens het bepaalde criteriamodel vertaald naar een eindwaarde per (sub)karakteristiek. Als het HIS aan de gestelde normen van alle checklists voldoet, is er ‘volledig voldaan’ aan de eisen. Als het HIS op geen van de checklists het deelresultaat ‘voldoet niet aan de acceptatiecriteria’ behaalt en op maximaal drie van de afzonderlijke toetsingsonderdelen het deelresultaat ‘voldoet voorwaardelijk aan de acceptatiecriteria’ dan is er sprake van voorwaardelijk voldaan. In de overige gevallen wordt er niet voldaan (WCIA, 1999).

De vier binnen de HIS-beoordelingsmethode onderkende activiteiten worden in tabel 5.1 op de ISO 14598-activiteiten afgebeeld.

Tabel 5.1 Vergelijking tussen de binnen de HIS-beoordelingsmethode uitgevoerde activiteiten en de door ISO 14598 onderkende activiteiten.

ISO 14598	HIS beoordeling
Bepaal doel van beoordeling	Bepalen van kwaliteitsprofiel
Identificeer producttypen	
Specificeer kwaliteitsmodel	Bepalen van kwaliteitsprofiel
Selecteer metrieken	Keuze van technieken, Construeren van checklists
Bepalen van kwalificatieniveaus voor de metrieken	Construeren van checklists
Bepalen criteria voor de beoordeling	Construeren van checklists
Formuleren van beoordelingsplan	
Uitvoeren van de metingen	Beoordeling uitvoeren
Vergelijken met criteria	Beoordeling uitvoeren
Vaststellen van de resultaten	Beoordeling uitvoeren

Deze afbeelding laat zien dat de HIS-beoordelingsmethode aan een groot deel van de ISO-activiteiten aandacht besteed. Uit het voorgaande blijkt dat activiteiten als ‘construeer checklists’ en ‘keuze technieken’ sterk vanuit het opgestelde kwaliteitsprofiel zijn opgezet. Omdat dit profiel wordt opgevat als operationalisering van de doelstelling wordt de uitvoering van deze activiteiten beïnvloed door het geformuleerde doel, hiermee is er sprake van aansturing.

Deze aansturing is niet volledig, er is sprake van beperkte aansturing. Illustratief hiervoor is de gang van zaken rondom het construeren van de checklists. Door de enorme hoeveelheid WCIA-eisen was er sprake van een continu wikken en wegen rondom het opnemen van vragen in de checklist. De kwaliteitskarakteristieken, genoemd in het kwaliteitsprofiel, gaven te weinig informatie om een dergelijke keuze op te baseren. Het opstellen van de checklist leunde dan ook zwaar op het inzicht en ervaring van de NHG- en KEMA-experts. Gezien de omstandigheden misschien een goede keuze, maar vanuit het perspectief van doelgerichte aansturing had men in feite behoefte aan meer ondersteuning.

Wat betreft de subaspecten middelenallocatie en voortgangsbewaking constateren we dat er wel sprake is van aansturing. Voor elk van de activiteiten is redelijk expliciet gemaakt welke middelen gebruikt worden. Zo worden de items voor de checklists tijdens activiteit ‘construeren checklists’ geselecteerd uit bestaande checklists, het WCIA-referentiemodel en ISO 12119 (1994) standaard. En voor de activiteit ‘uitvoeren beoordeling’ is bekend welke checklists toegepast moeten worden. Ook is bekend welke personen –welke functies– bij de beoordeling betrokken zijn. Wat betreft de voortgangsbewaking constateren we dat er per activiteit duidelijk is vastgesteld hoeveel tijd aan de verschillende activiteiten mag worden besteed. Dit geldt voor zowel het opstellen van kwaliteitsprofiel en checklists, als voor het uitvoeren van de beoordeling.

5.7 Ervaringen met het afwegen van doel en middelen

Bij het afwegen spelen zowel doel als middelen een rol. In paragraaf 5.5 is al ingegaan op de doelformulering. Er is daar aangegeven dat het doel van de beoordeling wordt uitgewerkt tot een kwaliteitsprofiel en dat een deel van het beoordelingsdoel –namelijk de verkorting van de doorlooptijd, en daarmee kostenreductie– is verdwenen De afweging wordt daarmee beperkt tot het afwegen op basis van kwaliteitseisen. Dit vormt de basis van de kritiek dat de beoordeling 'te zwaar' is en de twijfel achteraf bij NHG én KEMA over de geschiktheid van het gekozen instrumentarium.

Bij de afweging op basis van kwaliteitseisen speelt het kwaliteitsprofiel een belangrijke rol. Dit profiel geeft immers aan welke kwaliteitskarakteristieken belangrijk en welke minder belangrijk geacht worden. Vervolgens moeten geschikte beoordelingstechnieken worden gekozen. Dit onderwerp komt tijdens de HIS-beoordelingsmethode aan de orde met het uitvoeren van de activiteit 'keuze technieken'. Hier wordt bepaald welke statische en dynamische technieken worden toegepast. Tijdens deze techniekselectie speelt de twijfel over checklists een rol. Checklists brengen een validiteitsprobleem met zich mee; zie probleem in paragraaf 5.4. Er wordt toch overwogen om deze techniek te gebruiken omdat er niet een betere techniek beschikbaar is

De afweging betreft alleen de beoordelingstechnieken. Over de personen die bij de beoordeling betrokken dienen te zijn wordt niet gesproken. De afweging beperkt zich daarmee: er wordt gesproken over technieken, terwijl de toepassing van de techniek (checklists) juist mensen vereist.

Na de techniekkeuze vindt een tweede afweging plaats. De gekozen technieken (checklists) moesten namelijk worden uitgewerkt. Hiervoor werden bestaande checklists als uitgangspunt genomen. Deze bestaande kennis werd geraadpleegd omdat hiermee ervaring was opgedaan tijdens het beoordelen van andere producten. Men wist op deze manier wat de betreffende checklist(s) moesten meten: wat wordt er onder de verschillende kwaliteitskarakteristieken verstaan? In de vorige paragraaf is hierover opgemerkt dat het kwaliteitsprofiel, hiervoor onvoldoende basis biedt. Bij het overnemen van items uit bestaande speelt het inzicht van de NHG- en KEMA-experts een grotere rol. Het kwaliteitsprofiel vormde dus wel een basis tijdens de afweging, maar was te beperkt is om een gefundeerde afweging te maken.

5.8 Ervaringen met terugkoppeling en bijsturen van proces

Tijdens de HIS-beoordelingsmethode is er teruggekoppeld vanuit de activiteiten 'bepalen kwaliteitsprofiel', 'kiezen van technieken' en 'construeren van checklists'. Na het uitvoeren van deze activiteiten is er namelijk overleg geweest tussen de ontwerper van het beoordelingsproces en de opdrachtgever over de bereikte resultaten. Ook zijn er vervolgens wijzigingen en voorstellen tot verbetering doorgevoerd. Zo is het kwaliteitsprofiel na de

eerste vaststelling voorgelegd aan de NHG en heeft deze organisatie een aantal beoordelingsniveaus gewijzigd. Ook bij de twee andere activiteiten is sprake van dergelijke feedback: de checklists zijn bijgesteld. Deze terugkoppelingen en het vervolgens bijstellen is er steeds op gericht om de NHG –de opdrachtgever– bij de beoordeling te betrekken en tevreden te stellen. Over de vierde activiteit, 'uitvoeren beoordeling', kunnen we niets zeggen over terugkoppeling omdat deze niet tijdens de casestudie bestudeerd is.

Voor alle activiteiten, dus ook de drie activiteiten die gevolgd zijn door een review, wordt in de HIS-beoordelingsmethode niets opgemerkt over terugkoppeling en bijstelling. De output van elk van de activiteiten loopt naar een opvolger, maar er wordt niets gezegd over een vergelijking met het doel van beoordeling, ofwel terugkoppeling. In de praktijk van de HIS-beoordelingsmethode onderkennen we dus wel acties die lijken op feedback, maar feedback is niet structureel in de HIS-beoordelingsmethode opgenomen. Feedback wordt dus niet meegenomen en daardoor bestaat het gevaar dat het een sluitpost is: het wordt uitgevoerd als er 'toevallig' aan wordt gedacht. Zo is tijdens de HIS-beoordelingsmethode veel aandacht besteed aan het reviewen van de kwaliteitsprofielen en minder aandacht aan de review van checklists. Hierbij speelt nog een tweede reden voor het gebrek aan terugkoppeling, namelijk: het ontbreken van een kader om de resultaten van de activiteiten te vergelijken met het doel van beoordeling.

5.9 Samenvatting

De HIS casestudie laat zien dat de in hoofdstuk 3 geconstateerde problemen ten aanzien van de besturing van het beoordelingsproces in de praktijk relevant zijn. Dit geldt in deze beoordeling vooral voor de problematiek rondom het balanceren van doel en middelen en de tijdens het proces uit te voeren terugkoppeling en bijstelling.

Voor het formuleren van doel van beoordeling wordt het kwaliteitsprofiel gebruikt als operationalisering van het abstracte doel. Daarmee gaat de operationalisering alleen in op de eisen aan het product en komt een ander doel dat ten grondslag lag aan de beoordeling, namelijk het verkorten van de doorlooptijd, niet verder aan de orde. Het beoordelingsdoel is daarmee te beperkt geformuleerd.

Wat betreft de aansturing is geconstateerd dat er sprake is van beperkte aansturing: er vindt wel enige aansturing plaats, maar er wordt ook op een aantal punten gebrekkige aansturing aangetroffen.

De afweging van doel en middelen wordt onvoldoende expliciet uitgevoerd. Er wordt weliswaar een kwaliteitsprofiel gebruikt dat het doel van de beoordeling (deels) verwoord en waarmee de technieken op logische wijze worden geselecteerd. Dat dit resulteert in de verzameling –omvangrijke– checklists is echter minder duidelijk. Het betreft dan niet zozeer

de keuze voor checklists. Dat er checklists worden toegepast is te verklaren omdat dit voor het beoordelen van een HIS een adequate techniek is. Dat er hoge kosten, vooral met het herhaald toepassen van deze techniek zijn gemoeid, lijkt minder te zijn afgewogen. De afweging heeft zich daarmee maar op een deel van het aspect ‘kwaliteit’ en in het geheel niet op het aspect ‘kwantiteit’ gericht.

Terugkoppeling vindt plaats naar kwaliteitsprofiel en ook de checklists worden gereviewed. Deze terugkoppeling vindt echter onvoldoende gestructureerd en onvoldoende expliciet plaats. Daardoor wordt de terugkoppelingsinformatie niet gebruikt om het beoordelingsproces of activiteiten bij te sturen. De resultaten van de activiteiten worden tijdens een review alleen goed- of afgekeurd. Een belangrijk manco daarbij is dat het terugkoppelingsprincipe niet in de processtructuur is onderkend.

Deze casestudie is gestart met het in kaart brengen van de partijen die bij de ontwikkeling van de HIS-beoordelingsmethoden betrokken zijn. Er is geconstateerd dat de verschillende partijen in verschillende mate en met verschillende belangen bij dit proces betrokken werden en dat dit van invloed is geweest op de wijze waarop zij het resultaat beschouwden. Dit geeft een goede illustratie van de invloed van het politieke krachtenveld op de besturing van de beoordeling.

6. Een ontwerp voor doelgericht beoordelen

6.1 Inleiding

In de voorgaande hoofdstukken zijn vier belangrijke problemen ten aanzien van de besturing van een beoordelingproces geïdentificeerd en toegelicht. Wij vatten deze problemen nog eens samen.

Het eerste probleem betreft de doelformulering. In de praktijk wordt het doel van een beoordeling onvoldoende expliciet gemaakt. Zo wordt een doel vaak in te algemene termen verwoord en onvoldoende doorvertaald in termen van meten. Daarnaast wordt er vaak van slechts één doel uitgegaan, terwijl er meestal sprake is van verschillende, soms tegenstrijdige doelen. Een slechte doelformulering heeft nadelige gevolgen voor een beoordelingsproces. Doelen gaan door elkaar lopen en er ontstaan misverstanden en soms zelfs ruzies over de vraag waar de beoordeling ook al weer op was gericht.

Het tweede probleem betreft de aansturing van een beoordelingsproces. In de cases is naar voren gekomen dat het inrichten van het beoordelingsproces weinig doelgericht plaatsvindt. Soms wordt in het geheel geen proces onderkend, waardoor ook geen aansturing kan plaatsvinden. Soms zijn de activiteiten en de relaties daartussen wel gedefinieerd in een processtructuur, maar sluit de uitvoering ervan niet aan op het doel van beoordeling. Hierdoor ontstaan beoordelingen waarbij bijvoorbeeld metingen worden uitgevoerd zonder rekening te houden met het te bereiken doel van beoordeling.

Het derde probleem betreft de afstemming van doel en middelen. In de cases kwam naar voren dat er veelal geen afweging plaatsvindt, waardoor er onvoldoende rekening wordt gehouden met de vereiste inzet van geld, tijd en menskracht. Dit kan leiden tot beoordelingen die veel tijd en geld kosten zonder veel meerwaarde voor het uiteindelijk te bereiken resultaat. Dit is een bron van teleurstelling en frustratie.

Het vierde probleem betreft de terugkoppeling tijdens en de bijstelling van een beoordelingsproces. Het beoordelen van softwareproducten is geen 'rechttoe, rechtaan' proces, maar vereist voortdurende terugkoppeling naar eerder uitgevoerde stappen. Zo moet men zich tijdens een beoordelingsproces kritisch afvragen of gekozen normwaarden wel voldoende zijn toegespitst op het product dat beoordeeld wordt, of de juiste metrieken worden gehanteerd, of de meetresultaten daadwerkelijk toegevoegde waarde hebben, enzovoorts. De praktijk laat zien dat er van dergelijke reflecties weinig terechtkomt. Er is soms sprake van het reviewen van beoordelingsresultaten, maar een terugkoppeling naar voorgaande activiteiten vindt nauwelijks plaats. Ook in de literatuur vinden we weinig aanknopingspunten voor dergelijke feedback.

In dit hoofdstuk wordt een ontwerp opgesteld waarmee we deze problemen willen ondervangen. Dit ontwerp, ook wel conceptueel beoordelingsmodel genoemd, duiden we aan als ‘doelgericht beoordelen’.

6.2 Doelformulering

We gaan eerst in op het probleem dat doelen onvoldoende helder worden geformuleerd. De meest rationele aanbeveling zou luiden dat dit dan maar beter moet gebeuren door doelen scherp te formuleren. Een dergelijke aanbeveling doet echter geen recht aan de hoge moeilijkheidsgraad van het bepalen en beschrijven van doelen. Dat is geen triviale exercitie, maar een uiterst ingewikkeld proces waarin vaak diverse partijen een rol spelen. Dit geldt overigens niet alleen voor doelformuleringen van beoordelingsprocessen, maar ook voor vele andere terreinen, bijvoorbeeld op het gebied van vaststelling en evaluatie van overheidsbeleid (Hoogerwerf, 1989).

Op basis van de casestudies en interviews met mensen uit de beoordelingspraktijk komen we tot de volgende drie kernproblemen ten aanzien van het formuleren van doelen voor softwarebeoordelingen, namelijk:

1. vage of nietszeggende doelstellingen,
2. onmeetbare doelstellingen,
3. veranderende doelstellingen.

Vage of nietszeggende doelstellingen – een probleem dat zich vaak voordoet bij het formuleren van doelstellingen is dat doelen op een te hoog abstractieniveau worden geformuleerd waardoor het voor betrokkenen onhelder is wat nu precies beoogd wordt en tot welke acties dit zou moeten leiden. Dit is vaak het gevolg van het streven naar consensus om te voorkomen dat er vanuit de verschillende partijen verschillende doelen worden geformuleerd. Het spanningsveld rondom de conflicterende belangen en perspectieven wordt hiermee verhuld. De werkelijkheid laat zien dat dit een schijnoplossing is en dat partijen alsnog hun eigen doelen willen verwezenlijken: is het niet goedschiks, dan maar kwaadschiks. In plaats van een houding van ‘geven en nemen’ onstaat er als reactie op het onder de tafel schuiven van conflicterende doelstellingen een houding van het per sé vast houden aan de eigen uitgangspunten.

Onmeetbare doelstellingen – doelen zijn vaak onvoldoende doordacht wat betreft haalbaarheid en meetbaarheid. Er is vaak sprake van een algemene formulering, zonder dat er aanknopingspunten bestaan voor bijvoorbeeld het bepalen van kwaliteitskarakteristieken en daarbij passende metrieken. Gevolg is een onvoldoende koppeling tussen wat men wil bereiken met een beoordeling en de diverse beoordelingsactiviteiten.

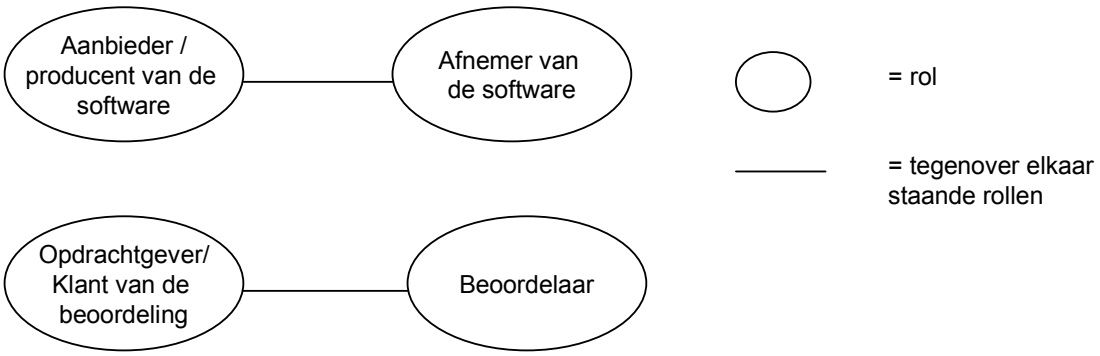
Veranderende doelstellingen – doelen van een beoordeling veranderen in de loop van de tijd, omdat het inzicht op de beoordeling van de betrokken personen verandert, omdat er alsnog andere partijen bij komen of omdat de omgeving van het beoordelingsproces verandert. Doelstellingen zijn daarmee vrijwel nooit constant maar veranderen tijdens een beoordelingsproces.

Om voorgaande kernproblemen rondom doelformulering aan te pakken dient er ons inziens aandacht worden besteed aan drie richtlijnen:
1. Identificeer partijen aan de hand van standaardrollen in het proces.
2. Operationaliseer het beoordelingsdoel door het als meetdoel met behulp van het measurement goal template te formuleren.
3. Voer versiebeheer uit met betrekking tot veranderende doelstellingen.

Identificeer partijen aan de hand van standaardrollen in het proces
Om doelen expliciet te kunnen formuleren is het belangrijk om te onderkennen dat er bij een beoordeling verschillende partijen betrokken zijn die elk hun eigen invulling van het te bereiken doel hebben. Een hulpmiddel om de partijen te identificeren is het onderkenen van rollen of functies in een beoordeling. In een beoordelingsproces kunnen tenminste vier rollen onderkend worden, namelijk:
- aanbieder (leverancier) of producent van de software,
- afnemer van de software, de klant van de aanbieder,
- beoordelaar, de beoordelende persoon of instantie,
- opdrachtgever van de beoordeling.

Niet alle rollen zullen in iedere beoordeling even prominent aanwezig zijn. Soms vallen de rollen samen en vervult een partij meerdere rollen. Zo namen de ontwikkelaars in de Omega-casestudie zowel de rol in van aanbieder van de software als die van de opdrachtgever van de beoordeling.

Van de vier rollen staan er telkens twee tegenover elkaar. De aannemer staat tegenover de afnemer van het product, de beoordelaar staat tegenover de opdrachtgever van de beoordeling. De rollen worden in onderstaande figuur weergegeven.



Figuur 6.1 Standaardrollen in een beoordeling

Door het onderkennen van deze rollen wordt duidelijk dat er tegengestelde posities in een beoordeling te vinden zijn. Dit maakt het gemakkelijker om doelen, problemen en meningen in een voorliggende beoordeling te herkennen en te plaatsen.

Voor het inventariseren van doelen zelf, zijn verschillende methoden bekend. Een voorbeeld is de 'probleem inventarisatie' in de bekende ISAC methode voor systeemontwikkeling van Lundeberg e.a. (1982). Zij leggen de nadruk op het inventariseren van de problemen van belanghebbenden en introduceren hiervoor een apart hulpmiddel: de belangengroeptabel. Dit is een beschrijving die wordt toegepast om de belangengroepen. te koppelen aan de problemen die zij ondervinden. Vanuit de onderkende belangen en problemen worden dan vervolgens doelen geformuleerd. Hierbij wordt verondersteld dat de belangen en problemen richting geven aan de diverse doelen van de partijen: de belanghebbenden zullen immers hun problemen opgelost willen zien.

Checkland (1981) komt met een vergelijkbare methode. Zijn aanbeveling is om aan het begin van elk systeemontwikkelingstraject alle betrokken partijen te inventariseren, alsmede hun problemen en wensen tot verbetering en hun (expliciete) doelen. Eén en ander wordt beschreven in een zogenaamde Rich Picture. Op deze manier tracht Checkland duidelijk te maken dat er verschillende belangen spelen bij een project en dat het wijs is daarmee expliciet rekening te houden qua aanpak (welke partijen wanneer aan het woord laten en raadplegen?), afbakening van systeemgrenzen (wat wel en wat niet aanpakken?) en oplossingsrichtingen (welke doelen wel en welke niet realiseren?). Door daarover duidelijk te communiceren, ontstaat er in elk geval duidelijkheid bij de partijen wat men wel en niet kan verwachten van een project. Daarmee voorkomt men fuzzyness en verwijten achteraf dat het project toch niet heeft opgeleverd wat men ervan had verwacht. Men zou dit kunnen samenvatten als het *managen van verwachtingspatronen*.

Operationaliseer beoordelingsdoel door het als meetdoel te formuleren
Het operationaliseren van doelen houdt in dat doelen geformuleerd worden in termen van de uit te voeren activiteiten en metingen. Het kwaliteitsprofiel dat tijdens de HIS-

beoordelingsmethode werd geformuleerd is hiervan een voorbeeld. Een bruikbaar concept voor het operationaliseren van doelen is het *measurement goal template* dat in studies over de Goal-Question-Metric-methode (GQM-methode) (Basili en Weiss, 1984), (Basili en Rombach, 1988) wordt gebruikt om meetdoelen te formuleren. Met dit template worden verscheidene zaken vastgelegd om de doelstelling uit te drukken in meetbare termen. Zaken die daarbij een rol spelen zijn:

- *Object* – dit betreft het onderwerp van de meting. Voor het verder indelen van objecten worden veelal de categorieën product, proces en (hulp)middelen gebruikt van Fenton en Pfleeger (1996).
- *Purpose* – het tweede element van de template betreft de *doelcategorie* waartoe het (meet)doel behoort. In de GQM-literatuur worden daarvoor een aantal categorieën gegeven, te weten: characterization, evaluation, prediction, improvement (Briand e.a., 1999a). Anderen noemen als doelcategorieën learning, control en improvement (van Solingen en Berghout, 1999). Beide indelingen zijn ons inziens te gebruiken om meetdoelen voor productbeoordelingen te expliciteren.
- Quality focus – dit derde element van het template betreft de vast te stellen eigenschappen van het object (Van Latum e.a., 1996). Voorbeelden zijn: cost, correctness, defect removal, changes, reliability. Voor het beoordelen van softwareproducten kan de indeling in kwaliteitskarakteristieken die het ISO-9126 kwaliteitsmodel geeft, worden gebruikt. Er kunnen ook meer gedetailleerde kwaliteitsmodellen worden gebruikt. Dit is aan de belanghebbenden, betrokken bij de beoordeling.
- *Viewpoint* – dit betreft het gezichtspunt van waaruit de beoordeling wordt uitgevoerd. Voorbeelden zijn metingen vanuit het perspectief van gebruiker, opdrachtgever, manager of ontwikkelaar. In dit licht dienen ook de door de ISO 14598 voorgestelde drie categorieën developer, acquirer en evaluator te worden gezien. Belangrijk is dat het gezichtspunt zo specifiek mogelijk wordt omschreven en dat de terminologie aansluit bij de belevingswereld van de betrokkenen. In het geval bijv de term “engineers” als een goede term wordt ervaren door de betrokken belangengroep, dan dient deze term te worden gebruikt in plaats van bijvoorbeeld “ontwikkelaar” of een andere term.

Als het template volledig wordt ingevuld, dan levert dat een zin op waarmee een *meetdoel* zo expliciet als mogelijk wordt geformuleerd: ‘analyseer {object} om te {purpose}, waarbij gelet wordt op {quality focus} vanuit het gezichtspunt {viewpoint}’. Het template kan als volgt worden weergegeven middels een tabel.

Object	
Purpose	
Quality focus	
Viewpoint	

Figuur 6.2 Template voor het formuleren van meetdoelen

Voer versiebeheer van doelen uit

Het voorgaande ging over het formuleren van doelstellingen van een beoordelingsproces en over het duidelijk en in meetbare termen uitdrukken van doelstellingen. De winst daarvan is evident: minder ambiguïteit en daardoor minder discussies tijdens een beoordelingsproces, wat nu wel of niet met een beoordeling wordt beoogd. Dit kan stagnaties en voortdurende terugkoppelingen voorkomen. Het schept bovendien voor de betrokkenen een realistisch verwachtingspatroon waardoor teleurstellingen achteraf zo veel als mogelijk worden vermeden. We kunnen dit niet geheel uitsluiten omdat één van de kernproblemen juist is dat doelen veranderen.

Het kunnen omgaan met dergelijke ‘moving targets’ vergt bijzondere inspanningen met betrekking tot het aansturen van een proces, met betrekking tot doel-middel afwegingen en met betrekking tot terugkoppeling en bijstelling. Het is hierbij noodzakelijk om tijdens een beoordelingsproces na te gaan of de oorspronkelijk geformuleerde doelen nog steeds overeind staan of te constateren dat er andere of nieuwe doelen worden nagestreefd. Indien dat zo is dan ontkomt men er niet aan om pas op de plaats te maken en terug te koppelen naar voorgaande activiteiten om na te gaan of men nog wel op het goede spoor zit. Voor de doelformulering betekent dit dat het nodig is om bij te houden welke versie van doelstellingen actueel is. Dit kan door doelformulering van karakteriseringen te voorzien zoals: de partij, tijdstip en geldigheid van de doelformulering. Het op deze manier karakteriseren van doelformuleringen duiden we aan als versiebeheer. Met dergelijk versiebeheer kan te allen tijde worden bepaald welke doelen worden nagestreefd door de betrokken partijen.

6.3 Aansturing met een aangepaste processtructuur

In deze paragraaf gaan we in op het probleem ‘onvoldoende procesaansturing’. Verbetering hierin zoeken we in het aanpassen van de processtructuur. Uitgangspunt hierbij is dat het doel van de beoordeling geoperationaliseerd wordt tot het niveau waarop de activiteiten plaatsvinden. Bij een dergelijke operationalisering is altijd sprake van het uitwerken van een doelhiërarchie. Zo wordt vanuit een doel een verzameling subdoelen opgesteld, die op hun beurt weer uit doelen kunnen bestaan. Deze operationalisering wordt verder uitgewerkt tot op het activiteitenniveau.

Een beproefd mechanisme om een doel tot op het niveau van meten te operationaliseren is het *Goal-Question-Metric (GQM)-principe*. Dit principe stelt dat er pas doelgericht kan worden gemeten wanneer doelen vertaald worden in te beantwoorden vragen en deze weer in metrieken. Het principe beschrijft zo de overgang van doel naar meetactiviteit.

Het GQM-principe is gebaseerd op ideeën en ervaringen van Basili and Weiss (1984), Basili e.a., (1986) en Rombach en Basili (1987) met het meten aan software processen en producten. Het GQM-principe wordt in de literatuur vaak aangeduid als GQM-paradigma. In de loop der tijd hebben verschillende auteurs het principe gebruikt of werkten het verder uit (Park e.a., 1996), (Van Latum e.a., 1998), (Van Solingen en Berghout, 1999), (Briand e.a., 1999a).

Bij onze uitwerking van het aansturen van een beoordelingsproces gaan we uit van de processtructuur zoals beschreven in ISO 14598. Dit is immers de internationale standaard die voorschriften voor het beoordelingsproces geeft. Het is dus nuttig om juist hiervoor verbeteringen op te stellen i.p.v. verbeteringen in een processtructuur van een willekeurige beoordelingsmethode. We hebben ons al in hoofdstuk 3 op deze standaard gericht.

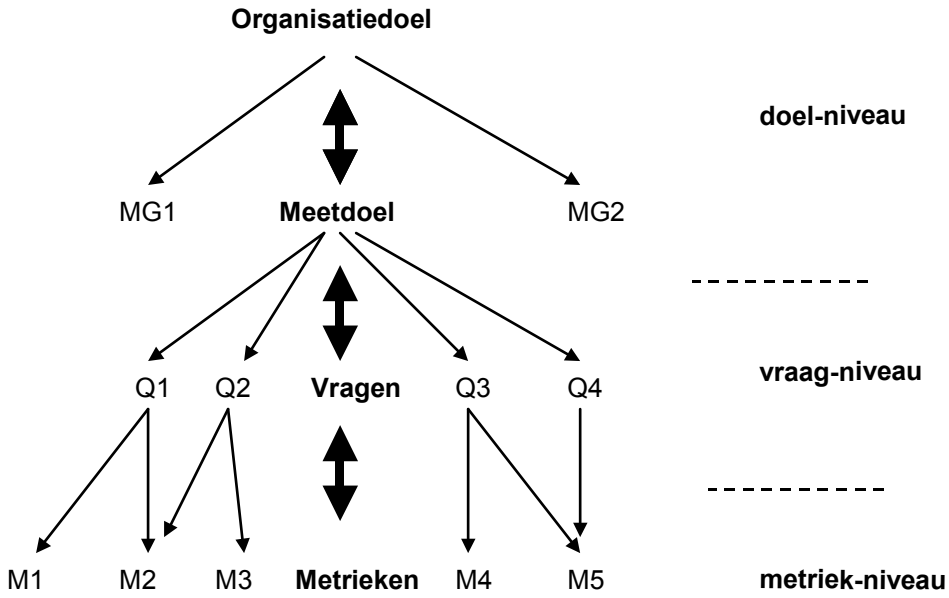
Uitgangspunten bij het aanpassen van de processtructuur
Tijdens het aanpassen van de processtructuur hanteren we twee uitgangspunten, namelijk, het expliciet onderkennen van doel-middel relaties en feedback in de processtructuur.

Doel-middel relaties in het GQM-principe – in het Goal Question Metric (GQM)-principe is sprake van twee doel-middel relaties, namelijk: de relatie doel-vraag en de relatie vraag-metriek. De Leeuw (1980) merkt over doel en middel op dat het twee kanten van dezelfde medaille zijn: ‘als je hard werkt om een boek door te worstelen kan dat gezien worden als middel. Je kunt dan zeggen dat je doel is om in een nieuwe gebied thuis raken. Het thuis raken in een nieuwe gebied kan op zijn beurt weer een middel zijn om van werkkring te veranderen’. In het GQM-principe geldt iets soortgelijks: het stellen van vragen is nodig om het doel te operationaliseren, terwijl metrieken nodig zijn om gestelde vragen te beantwoorden.

Een derde doel-middel relatie betreft de relatie bedrijfsdoel – meetdoel. Veel GQM-benaderingen starten vanuit een meetdoel. In paragraaf 6.2 is al aangegeven dat het dan om een expliciete omschrijving van het doel in meettermen gaat. Dit is wat anders dan het organisatiedoel (business goal). Hiermee wordt geformuleerd wat men als groep mensen in een organisatie wil bereiken. Het doel van een beoordeling in termen van meetdoelen zal altijd een subdoel zijn van een organisatiedoel. Op het niveau van een organisatiedoel gaat het om een bredere context dan de beoordeling van software alleen. Het voorgaande leidt ertoe dat we bij het doel van een beoordeling twee niveaus kunnen onderscheiden, namelijk dat van organisatiedoelen en dat van meetdoelen.

Op deze manier komen we tot drie doel-middel relaties die we toepassen om de ISO 14598 processtructuur te herstructureren, namelijk:
- organisatiedoel – meetdoel,
- meetdoel – vraag,
- vraag – metriek.

Grafisch geven we dit weer met de volgende boomstructuur. Deze is afgeleid van de GQM-boomstructuur uit (Van Latum e.a., 1996), (Van Solingen en Berghout, 1999).



Figuur 6.3 Doel-middel relaties in het GQM-principe

Feedback – naast doel-middel relaties dient er in de processtructuur expliciet rekening te worden gehouden met het tussentijds analyseren van de bereikte resultaten om zo een proces eventueel bij te sturen als daar aanleiding toe is. Deze terugkoppeling geschiedt in feite per activiteit en zal worden aangeduid met de term review: de resultaten worden geanalyseerd en besproken waarna zo nodig wordt bijgesteld.

Uitwerking
Hieronder wordt per activiteit beschreven hoe de processtructuur volgens ons zou moeten worden gewijzigd.

Bepaal organisatiedoel – tijdens deze activiteit stellen we de organisatiedoelen vast. Dit gebeurt voor zover deze doelen een relatie met de beoordeling hebben. Het gaat dus niet om het formuleren van een organisatiemissie in het algemeen maar om een inventarisatie van die organisatiedoelen die een direkte relatie hebben met het beoordelen van software. Het gaat dus om het vaststellen van het doel *van* de beoordeling in relatie tot de

organisatie(processen): ‘wat willen we met de beoordeling bereiken?’ , ‘wat zijn de wensen ten aanzien van de beoordeling?’ etcetera Hierbij staat centraal de betekenis en de impact van een softwareproduct voor de organisatie. Door hier van uit te gaan zal het doel van de beoordeling van dit product duidelijker worden.

Een hulpmiddel voor het verhelderen van de relatie organisatiedoel-meetdoel is het opstellen van een kwaliteitsprofiel van het product (zie hoofdstuk 3 en 5). Dit profiel beschrijft namelijk wat men in de organisatie waar het product wordt gebruikt, belangrijk vindt. Dit vormt tevens input voor de ‘quality focus’ tijdens het bepalen van meetdoelen. Om het kwaliteitsprofiel op te stellen kunnen we bijvoorbeeld de Space-Ufo methode gebruiken(Space-Ufo consortium, 1998). Deze methode gaat ervan uit dat softwareproduct kwaliteit wordt bepaald door drie factoren, namelijk:

- Bedrijfsproces waarin het product wordt gebruikt. Space-Ufo onderkent twee soorten softwareproducten: (administratieve) informatiesystemen en embedded software. Voor beide soorten software wordt een aparte invulling gegeven van de factor “bedrijfsproces”. Zo wordt voor een embedded product bij bedrijfsproces met name gekeken naar de functionaliteit die de markt verlangt van het product.
- Gebruiker – de verwachtingen die deze ten aanzien van het product heeft. Bij embedded producten zal het hierbij gaan om externe klanten of klantgroepen.
- Omgeving – waarin het product wordt toegepast.

De methode omvat een vragenlijst om vanuit deze drie factoren een kwaliteitsprofiel op te stellen. Voor een verdere uitwerking wordt verwezen naar de methode zelf (Space-Ufo consortium, 1998) en (Van Ekris, 1998).

Door tijdens het bepalen van een organisatiedoel te focussen op de rol van het softwareproduct voor de organisatie wordt in feite aandacht besteed aan wat in de ISO 14598-processtructuur wordt aangeduid met de activiteit ‘identificeer producttypen’. In onze aanpassing van de processtructuur wordt deze activiteit ondergebracht in de activiteit ‘bepaal organisatiedoel’.

Bepaal meetdoelen – tijdens deze activiteit worden de meetdoelen van een beoordeling vastgesteld. Deze meetdoelen moeten uiteraard aansluiten op het eerder geformuleerde organisatiedoel. In paragraaf 6.2 is aangegeven dat het ‘measurement goal template’ kan worden gevolgd om meetdoelen te formuleren.

Als er veel meetdoelen worden geformuleerd, dan kan het zijn dat het onmogelijk is om elk van de meetdoelen uit te werken tot metrieken. Immers, de hiervoor benodigde tijd, het beschikbare geld en de benodigde capaciteit kunnen ontbreken; zie bijvoorbeeld (Lamprecht en Weber, 1998). Tijdens het formuleren van meetdoelen moet hieraan dan ook aandacht worden besteed. Dit kan door het toekennen van prioriteiten aan de doelen. In het geval van

beperkte middelen, hetgeen bijna steeds het geval zal zijn, zullen alleen de meest belangrijke meetdoelen worden omgezet in concrete metrieken. Men kan immers niet alles meten tot in de meest verfijnde vorm, gegeven doelstellingen enerzijds en beschikbare middelen anderzijds.

Wat een belangrijke of minder belangrijke doelstelling is, wordt bepaald door de belanghebbenden bij de beoordeling. Zij zullen moeten aangeven wat wel en niet belangrijk is, lettende op het beschikbare budget in termen van geld, capaciteit en doorlooptijd.

Definieer vragen en hypotheses – het tweede niveau van het GQM-principe is het vraag-niveau. Uitgangspunt voor deze activiteit zijn de opgestelde meetdoelen. Tijdens deze activiteit worden te beantwoorden vragen geformuleerd waarmee wordt tegemoet gekomen aan de gestelde meetdoelen. Ook worden er hypotheses opgesteld. Het resultaat van de activiteit zijn vragen en hypotheses waarmee een relatie met het metriek-niveau kan worden gelegd. Een voorbeeld: stel dat het doel van een beoordeling is om te achterhalen of een bepaald softwareproduct wel of niet gebruiksvriendelijk is in de praktijk. Dit kan enorm belangrijk zijn in gevallen waar vele gebruikers betrokken zijn bij zo'n softwareproduct zonder dat de mogelijkheid bestaat om deze gebruikers uitvoerig te trainen en te instrueren. Vragen die men in dit geval beantwoord zou willen zien zijn onder meer de volgende:
- Hoe snel kan een doorsnee gebruiker zich het softwareproduct eigen maken (leertijd).
- Hoe duidelijk is de bijgevoegde documentatie en instructie (gebruikershandleiding).
- Hoe zelf verklarend is het product (helpfuncties).

Voor elke vraag kan men een hypothese formuleren ten aanzien van bijvoorbeeld het minimumniveau waaraan een product moet voldoen Een concrete meting moet dan kunnen aangeven of voldaan wordt aan dat gestelde minimumniveau.

Sommige GQM-auteurs spreken in plaats van hypothese over 'indicator' (Park e.a., 1996) of 'measurement concepts' (Briand e.a.,1999a). Ze geven daarmee aan dat met het vaststellen van de hypotheses keuzes ten aanzien van metrieken worden gemaakt of sterker: dat de metrieken eigenlijk dan al gedefinieerd worden. Wij gaan niet zo ver en zien het opstellen van hypotheses meer als het bepalen van meetgrenzen.

Een belangrijk aandachtspunt bij het formuleren van vragen is het niveau waarop de vragen worden geformuleerd (Van Solingen en Berghout, 1999). Vragen worden soms te abstract en soms te concreet geformuleerd. Als ze te abstract worden geformuleerd, dan is er een probleem om de vertaling naar metrieken te maken. Als ze te concreet zijn geformuleerd, dan is de koppeling naar doelen weer moeilijk.

Bij het formuleren van vragen speelt het begrip mentaal model een belangrijke rol. Een *mentaal model* is een model waarin de impliciete kennis van mensen in een organisatie over

het softwareproduct is vastgelegd. Deze personen hebben ervaring met de software en hebben daardoor ook een beeld van de kwaliteit ervan. In de Omega-casestudie zagen we hier een voorbeeld van toen de software engineers van het Omega-systeem formuleerden wat goed en slecht onderhoudbare modules zijn. Mentale modellen zijn per definitie impliciet en subjectief: ze zijn gekoppeld aan personen. Tijdens het opstellen van vragen en in het vervolg daarvan het opstellen van hypotheses wordt deze kennis expliciet gemaakt.

De activiteiten 'definieer vragen en hypotheses' en 'bepaal meetdoel' zijn gerelateerd aan de activiteit die door ISO 14598 wordt aangeduid als 'specificeer kwaliteitsmodel'. Er wordt immers mee vastgelegd wat de te beoordelen norm is. Door vragen te stellen wordt bepaald wat een organisatie belangrijk vindt qua karakteristieken van een softwareproduct. Elke vraag kan in principe worden vertaald naar een kwaliteits(sub)karakteristiek. Het stellen van vragen heeft als voordeel dat ze zijn geformuleerd in natuurlijke taal of spreektaal van de betrokken mensen. Door de betrokken partijen op deze manier te laten communiceren worden de mentale modellen geëxpliciteerd.

Bepaal metrieken en normwaarden – het derde niveau van het GQM-principe is het metriek-niveau. Uitgangspunt voor deze activiteit zijn de eerder opgestelde vragen en hypotheses.

In ISO 14598 wordt het begrip kwalificatieniveau onderkend. Omdat we de term normwaarde prefereren wordt deze term in het vervolg gehanteerd. In de GQM-methoden krijgt het begrip normwaarde (of kwalificatieniveau) niet expliciet aandacht middels een activiteit behalve bij wat eerder is gezegd over het opstellen van hypotheses. Het lijkt wel alsof men ervan uitgaat dat de normwaarden automatisch volgen als de vragen en metrieken zijn bepaald. Onze ervaringen tijdens de casestudies zijn anders. Het lijkt ons dan ook beter er wel aandacht aan te besteden. Vandaar dat het begrip expliciet in de processtructuur wordt onderkend. Het wordt echter niet als aparte activiteit geïdentificeerd, zoals in de ISO14598-processtructuur gebeurt. Metrieken en normwaarden moeten in samenhang met elkaar bepaald worden. Vandaar dat er sprake is van één activiteit 'bepaal metrieken en normwaarden'.

Bepaal criteriamodel – tijdens deze activiteit worden de criteria om de meetresultaten te interpreteren bepaald en geformuleerd in een model. In dit model wordt vastgelegd hoe men komt van meetwaarden, tot een beantwoording van de vragen en vervolgens tot het toetsen of aan het meetdoel is voldaan. Het model heeft daarmee betrekking op zowel doel-, vraag- als metriek-niveau.

Deze activiteit is gerelateerd aan wat in de ISO 14598 processtructuur wordt aangeduid als 'bepaal criteria voor beoordeling'. In de door ons vastgestelde activiteit is meer sprake van het vastleggen en formaliseren van deze criteria dan van het bepalen ervan. De verbanden

tussen meetdoel, vragen en metrieken, en daarmee de criteria, zijn immers al tijdens voorgaande activiteiten vastgesteld.

Formuleer beoordelingsplan – na de vaststelling hoe er gemeten gaat worden, is het zaak om het beoordelingsplan te formuleren. Dit plan bestaat naast het criteriamodel ook uit de toewijzing van middelen en een plan voor de voortgangsbewaking voor de uitvoering van de beoordeling. Zo wordt ook aangegeven met welke mensen en eventuele tools de beoordeling wordt uitgevoerd.

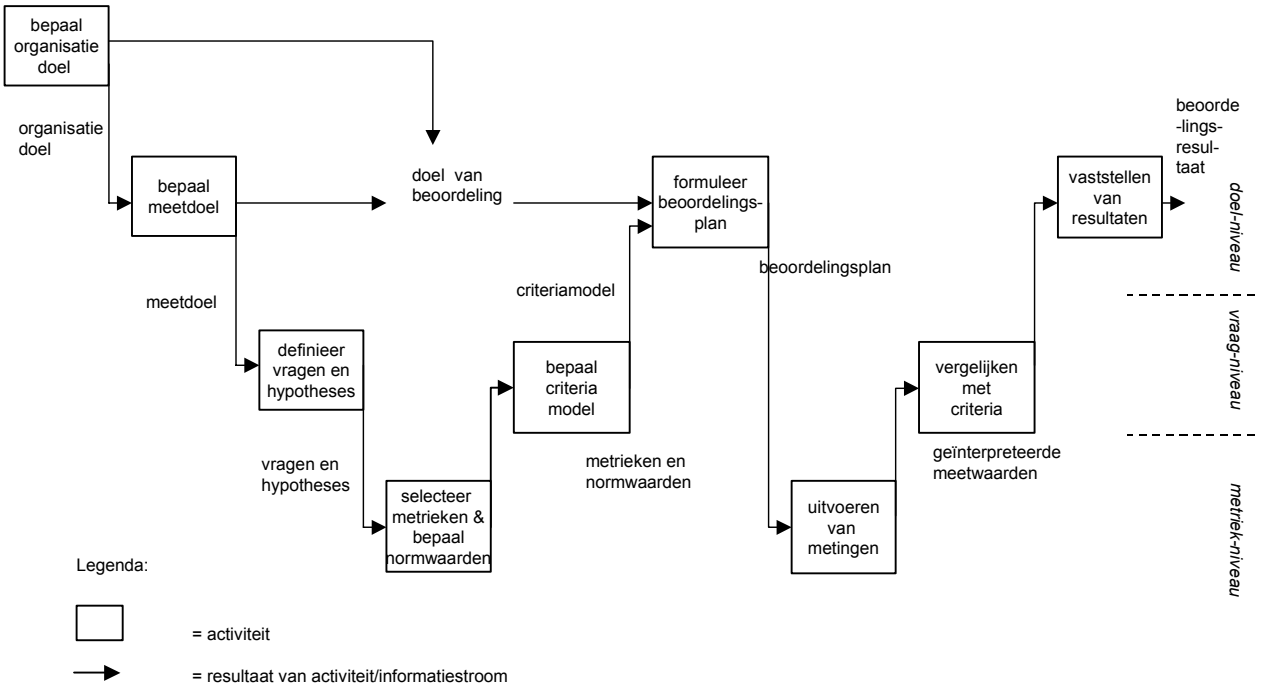
De input van de activiteit bestaat uit het zojuist opgestelde criteriamodel én het doel van beoordeling dat op organisatiedoel-niveau is opgesteld. Tijdens de activiteit vindt een afweging plaats van dit doel en van de middelen waarmee dit doel zal moeten worden bereikt. Er wordt nagegaan of de op de tot op dat moment opgestelde beoordeling aansluit op het doel van de beoordeling. Dit laatste wordt weergegeven als het criteriamodel. Ook wordt er bepaald of de opgestelde beoordeling gerealiseerd kan worden: zijn de mensen en middelen aanwezig? Als deze afweging positief uitpakt, dan wordt het beoordelingsplan opgesteld en kan met de uitvoering van de beoordeling worden gestart. Als de afweging negatief uitvalt dan moet het opstellen van meetdoelen, vragen en metrieken opnieuw worden gedaan. De meetdoelen, vragen en metrieken worden bijgesteld totdat het doel van beoordeling en het op dat moment geformuleerde beoordelingsplan voldoende op elkaar aansluiten.

Uitvoeren van metingen – tijdens deze activiteit worden de metrieken volgens het beoordelingsplan toegepast. Dit resulteert in actuele meetwaarden. Deze worden vergeleken met de normwaarden, wat resulteert in geïnterpreteerde meetwaarden. Deze activiteit wordt ook aangeduid in de ISO 14598-processtructuur.

Vergelijken met criteria – de geïnterpreteerde meetwaarden worden gebruikt om antwoord te geven op de gestelde vragen. De metingen worden immers uitgevoerd om vragen te beantwoorden. Hierbij maken we gebruik van het eerder opgestelde criteriamodel. Dit model definieert zoals we zagen, welke metrieken geselecteerd zijn voor de gestelde vragen en met welk gewicht ze meewegen in de totaalbeoordeling. De beantwoording van de vragen vindt plaats op vraag-niveau. Deze activiteit komt overeen met wat de ISO 14598-processtructuur aanduidt als ‘vergelijken met criteria’; daarom wordt deze term overgenomen.

Vaststellen van resultaten – tijdens deze activiteit wordt vastgesteld of het product voldoet aan de gestelde meetdoelen. Dit gebeurt op basis van de antwoorden die op de gestelde vragen gegeven zijn. De bepaling of het doel gehaald is, vindt plaats op doelniveau. De activiteit komt vrijwel overeen met wat de ISO 14598-processtructuur aanduidt als ‘vaststellen van resultaten’. We handhaven daarom deze omschrijving.

Met de activiteit ‘vaststellen van resultaten’ is in pricipe het einde van het beoordelingsproces bereikt. De processtructuur wordt in de volgende figuur beschreven.



Figuur 6.4 Aangepaste processtructuur

In de figuur zijn de drie niveaus van doel-middel-relaties duidelijk te herkennen: doel-, vraag- en metriekniveau. De eerste vier activiteiten hebben betrekking op met name het onderkennen van deze drie niveaus. Voor het doelniveau gaat het om de activiteiten: bepaal organisatiedoel en bepaal meetdoel. Voor het vraagniveau gaat het om: bepaal meetdoel, definieer vragen en hypotheses. Tenslotte gaat het op metriekniveau om: definieer vragen en hypotheses, selecteer metrieken & bepaal normwaarden.

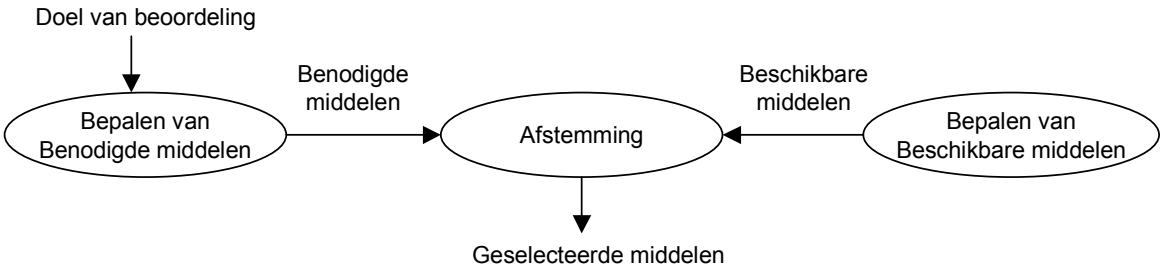
De figuur toont naast de activiteiten ook de relaties tussen de activiteiten. De meeste relaties hebben betrekking op de volgorde waarin de activiteiten worden uitgevoerd. Zo volgt de activiteit ‘formuleer beoordelingsplan’ op de activiteit ‘bepaal criteria model’.

6.4 Afweging door het expliciteren van doel en middelen

In deze paragraaf wordt ingegaan op de problematiek ‘onvoldoende afwegen van doel en middelen’. De verbetering wordt gezocht in het explicieter beschrijven van doel en middelen en het onderkennen van ‘afwegingsmomenten’ in de processtructuur. Om tot deze verbeteringen te komen, onderkennen we eerst drie fasen bij het afwegen, namelijk:

- Het bepalen van de kwaliteit en kwantiteit van de benodigde middelen.
- Het bepalen van de beschikbare middelen.
- Het afstemmen van de benodigde en beschikbare middelen.

In de eerste fase wordt vastgesteld welke kwantiteit en kwaliteit van de middelen vereist wordt om het doel in optima forma te bereiken. Vervolgens wordt vastgesteld wat de aanwezige middelen zijn wat betreft kwantiteit en kwaliteit. Tenslotte worden de benodigde en beschikbare middelen op elkaar afgestemd. De volgende figuur geeft de stappen in het afwegen weer.



Figuur 6.5 Drie stappen tijdens de afweging

Bepalen van de kwaliteit en kwantiteit van de benodigde middelen
Bij het bepalen van de benodigde middelen wordt uitgegaan van het doel van de beoordeling. Dit is immers de leidraad voor de procesuitvoering. In de casestudies hebben we gezien dat dergelijke doelen vaak niet expliciet en operationeel worden geformuleerd. Dat maakt dat er nauwelijks aanknopingspunten bestaan voor een afweging van doelen en benodigde middelen. Het verbeteringsvoorstel voor het formuleren van doelen in paragraaf 6.2 is dan ook een stap voorwaarts om tot een betere middelenafweging te komen. Het measurement goal template dat gebruikt wordt voor het operationaliseren van meetdoelen omvat drie elementen die geschikte eigenschappen zijn om de benodigde kwaliteit van de middelen te beschrijven. Het betreft: ‘object’, ‘quality focus’ en ‘viewpoint’. Met deze elementen wordt immers uitgedrukt wat het onderwerp van de beoordeling is en aan welke kwaliteitskarakteristieken het beoordelingsproces –en daarmee de benodigde inzet aan middelen– aandacht moet besteden. Het vierde element viewpoint geeft aan vanuit welk perspectief dat moet gebeuren. Hiermee wordt gedefinieerd welke mensen betrokken zijn bij een beoordeling.

Het measurement goal template zegt hier niets over de kwantiteit van de middelen. Het is daarom nuttig om het template aan te vullen met een vijfde element zijnde de kosten en de tijd die aan de beoordeling mogen worden besteed.

Bepalen van de kwantiteit en kwaliteit van de beschikbare middelen
Middelen kunnen variëren van de mensen die tijdens het proces worden ingezet tot een methode om een kwaliteitsprofiel op te stellen of een specifieke codemetriek. De beschikbaarheid van deze middelen hangt af van de ervaring die de organisatie met de

middelen heeft opgedaan en van de mensen die aanwezig zijn om deze middelen toe te passen.

De casestudies en de analyse van de beoordelingsmethoden bevestigen dat er vaak weinig bekend is over de kwaliteit en kwantiteit van de middelen. Weliswaar vindt men vele beschrijvingen van bijvoorbeeld beschikbare metrieken maar daarbij blijft in het ongewisse welke ondersteuning beschikbaar is of wat het toepassen van een metriek kost in termen van geld en tijd. De tweede verbetering van het afwegen wordt dan ook gezocht in het expliciteren van de kwaliteit en kwantiteit van de beschikbare middelen. Hieronder besteden we aandacht aan twee soorten hulpmiddelen, namelijk: mensen en metrieken.

Mensen
Een belangrijk middel dat wordt ingezet tijdens een beoordelingsproces zijn mensen. Hierbij kan het gaan om de beoordelaars die het beoordelingsplan opstellen, de metingen uitvoeren en een eindrapport opstellen. Een andere belangrijke groep zijn de gebruikers van het systeem die tijdens een beoordeling naar hun mening wordt gevraagd. Een derde groep betreft de partijen die de doelstellingen opstellen (zie paragraaf 6.2). Elk van deze groepen ondersteunt één of meer beoordelingsactiviteiten. Bij het afwegen van middelen zal er gekeken moeten worden naar beschikbaarheid van alle betrokken partijen in termen van kwantiteiten en kwaliteiten. Kwaliteit betreft dan onderwerpen als vaardigheid of kennisniveau van de mensen. Kwantiteit betreft dan zaken als beschikbaarheid en uurtarief van mensen.

Metrieken
Zoals eerder opgemerkt bestaat er veel literatuur over metrieken bij softwarebeoordelingen maar het schort aan ervaringsgegevens met deze metrieken in termen van benodigde inzet aan middelen en resultaat. Dat wordt met name veroorzaakt door het feit dat men blijkbaar geen gegevens verzamelt tijdens beoordelingsprocessen die later weer gebruikt kunnen worden als ‘ervaringsbank’.

Van de bestudeerde beoordelingsmethoden besteedt de Scope-methode het meeste aandacht aan het bijhouden van ervaringsinformatie over metrieken (zie hoofdstuk 3). De methode onderkent namelijk evaluatiemodules die geordend zijn naar kwaliteitskarakteristieken en beoordelingsniveaus. Hiermee wordt een eerste indruk gegeven van de ‘kwaliteit’ van het middel. Over kosten en benodigde tijd van de modules (de kwantiteit) zegt de methode echter niets.

In navolging van de Scope-methode stellen we voor om ervaringsgegevens over productbeoordelingen en daarbij gehanteerde kwaliteitskarakteristieken bij te houden. Van een metriek dient bekend te zijn op welk object het betrekking heeft en vanuit welk perspectief er gemeten wordt. Ervaringsgegevens met betrekking tot karakteristiek, object en

perspectief zijn alle van belang omdat deze attributen raken aan het meetdoel en er zo een basis is voor het afwegen.

Het concept 'beoordelingsniveau' dat de Scope-methode gebruikt is ook interessant om uitspraken te doen over de inzet van middelen. In het Scope-project is een indeling gemaakt waarbij beoordelingstechnieken, zoals checklists en statisch testen, zijn gerangschikt naar vier beoordelingsniveaus: A, B, C en D (Robert, 1994). Deze indeling is weliswaar erg algemeen en geeft alleen aan tot in welke mate van detail en precisie men een bepaalde meting wil en kan uitvoeren. We stellen voor om niet alleen complete technieken, in Scope de evaluatie modules, op te nemen, maar ook ervaringsinformatie over 'losse' metrieken te verzamelen. Op deze manier kan ook informatie over codemetrieken worden verzameld, waarna het mogelijk wordt om tijdens selectie combinaties van codemetrieken en checklist items voor te stellen. Het verzamelen van zowel checklist items als beschrijvingen van codemetrieken levert een breder scala aan metrieken dan een gegevensbanken met alleen metrieken, zoals (Zuse, 1998) of alleen checklists, zoals (Robert, 1994).

Een kritiek op de beoordelingsniveaus zoals Scope deze toepast, is dat de criteria voor het het kiezen uit de verschillende niveaus onduidelijk zijn en de schalen van de niveaus arbitrair (Scope IPSE, 1994), (Rae e.a., 1995), (van Uittregt, 1998). Ondanks deze kritiek denken we dat het idee achter beoordelingsniveaus belangrijk is voor het kwalificeren van metrieken. Het moet dan gaan om de meettechnische eigenschappen van de metrieken, i.c. vragenlijsten. Onder deze eigenschappen verstaan we de 'reproduceerbaarheid' en de 'validiteit' van metrieken. Het ware nuttig daaromtrent empirisch materiaal vast te leggen om voor toekomstige beoordelingen een inschatting te kunnen maken over het nut van dergelijke metrieken en de vereiste middelen daarbij.

Uitwerking: metriekendatabase

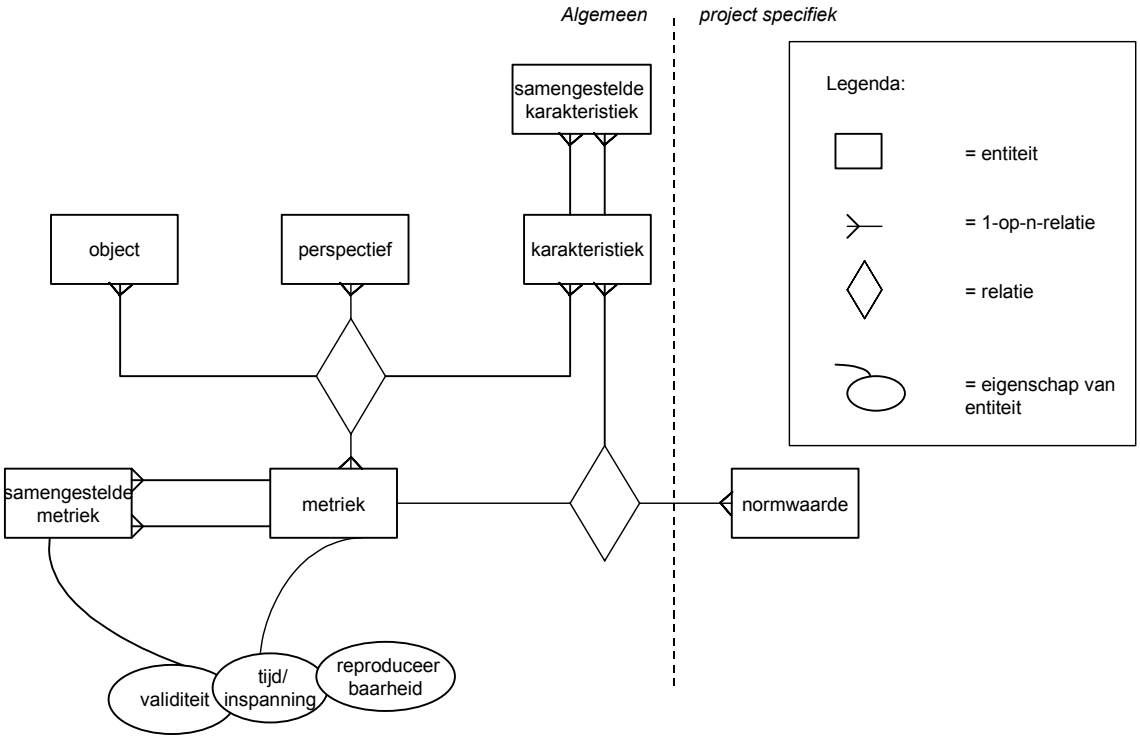
Om informatie over de kwaliteit en kwantiteit van metrieken te verzamelen is tijdens dit onderzoek een metrieken database ontwikkeld. Hierbij is uitgegaan van de voorgaande overwegingen om metrieken te expliciteren. Dit houdt in dat er attributen aan metrieken worden toegekend, die zijn gelieerd aan de elementen van het measurement goal template, namelijk:

- Object – dit is het onderwerp van beoordeling, bijvoorbeeld code of documentatie. Het betreft het element 'object' uit de template.
- Kwaliteitskarakteristiek – dit betreft het element 'quality focus' uit de template. De karakteristiek kan in principe elke kwaliteits(sub)karakteristiek zijn die er te bedenken valt. Een ordening volgens ISO 9126 is mogelijk. In de uitwerking van de database is gekozen voor beperking tot de karakteristieken rondom Maintainability, zoals Analysability en Changeability.
- Perspectief – dit betreft het gezichtspunt 'viewpoint' uit de template- van waaruit de beoordeling plaatsvindt.

Daarnaast zijn de attributen validiteit en reproduceerbaarheid van de metriek, zie hiervoor toegevoegd. Met betrekking tot de kwantiteit is het attribuut tijd/inspanning opgenomen.

Behalve deze attributen is overwogen om nog meer attributen toe te voegen om metrieken te identificeren. De literatuur rondom metrieken geeft vele mogelijke attributen (Oman e.a., 1991, 1992), (Baumert en McWhinney, 1982), (Squid consortium, 1996) en (Zuse, 1998). Deze attributen kunnen wellicht ook worden gebruikt om zo nog meer gegevens over de kwaliteit van metrieken in een database op te slaan om vervolgens in een nieuwe beoordelingssituatie metrieken te ontsluiten. Er is echter voor gekozen om over een beperkte set attributen gegevens bij te houden. Bestaande datamodellen, zoals (Squid consortium, 1996), (Basili e.a., 1986), (Basili en Rombach, 1987) laten zien dat er met meerdere attributen al snel te veel gegevens moeten worden bijhouden. Het gevaar is dan dat het bijhouden van gegevens verwaterd waardoor er maar over een beperkt aantal gevallen informatie beschikbaar is. In het geval van Squid zijn gegevens over 1 project opgenomen. Bij het ontwikkelen van de metriekendatabase zijn daarom een paar extra attributen toegevoegd, namelijk: soort meting (indirect versus direct en extern versus intern) en moment in levenscyclus van het product waarop de beoordeling plaatsvindt (ontwerp, codering, implementatie, onderhoud).

Het bovenstaande leidde tot een datamodel dat is geïmplementeerd in een database (Punter, 1998a). Onderstaande figuur toont het datamodel als Entiteit Relatie diagram.



Figuur 6.6 Datamodel metriekendatabase

Een deel van de metriekattributen is gemodelleerd als entiteit, andere als attribuut/eigenschap. De centrale entiteit is metriek. Metrieken hebben een complexe relatie met zowel het object waarop het van toepassing is, het perspectief van waaruit het wordt toegepast en de eigenschap waarover het een waarde moet genereren. In hoofdstuk 2 hebben we gezien dat metrieken nogal eens worden samengesteld uit andere: indirecte metriek. Om dit te modelleren is een vijfde entiteit opgenomen: samengestelde metriek. Naast metriek is ook bij (kwaliteits)karakteristiek nogal eens sprake van een samenhangend geheel van verschillende karakteristieken. Om deze hiërarchie van kwaliteitskarakteristieken te modelleren (denk aan bijvoorbeeld ISO 9126) is een extra entiteit toegevoegd, namelijk ‘samengestelde karakteristiek’.

Rechts in het diagram is de entiteit ‘normwaarde’ opgenomen. Door hierover gegevens bij te houden kunnen voor een concrete beoordelingssituatie suggesties voor dergelijke waarden worden gepresenteerd. Vandaar dat normwaarde is gemodelleerd in de metriekendatabase. Normwaarde betreft projectspecifieke informatie. Hiervoor is een onderscheid gemaakt tussen projectspecifieke kennis en algemene kennis. Dit is gebaseerd op de bevindingen van Van Genuchten (1991). Bij het meten aan software(projecten) wordt veel projectspecifieke kennis opgedaan. Dit moet worden gescheiden van meer algemene –en daarmee herbruikbare- kennis.

Normwaarde is in het onderstaande model eenvoudig als entiteit gemodelleerd. Hierachter schuilt een complexere modelleringsproblematiek waarbij kan worden overwogen om dit te modelleren middels regels (Steels, 1992), (Qiu, 1995). In dit proefschrift wordt hier niet verder op ingegaan.

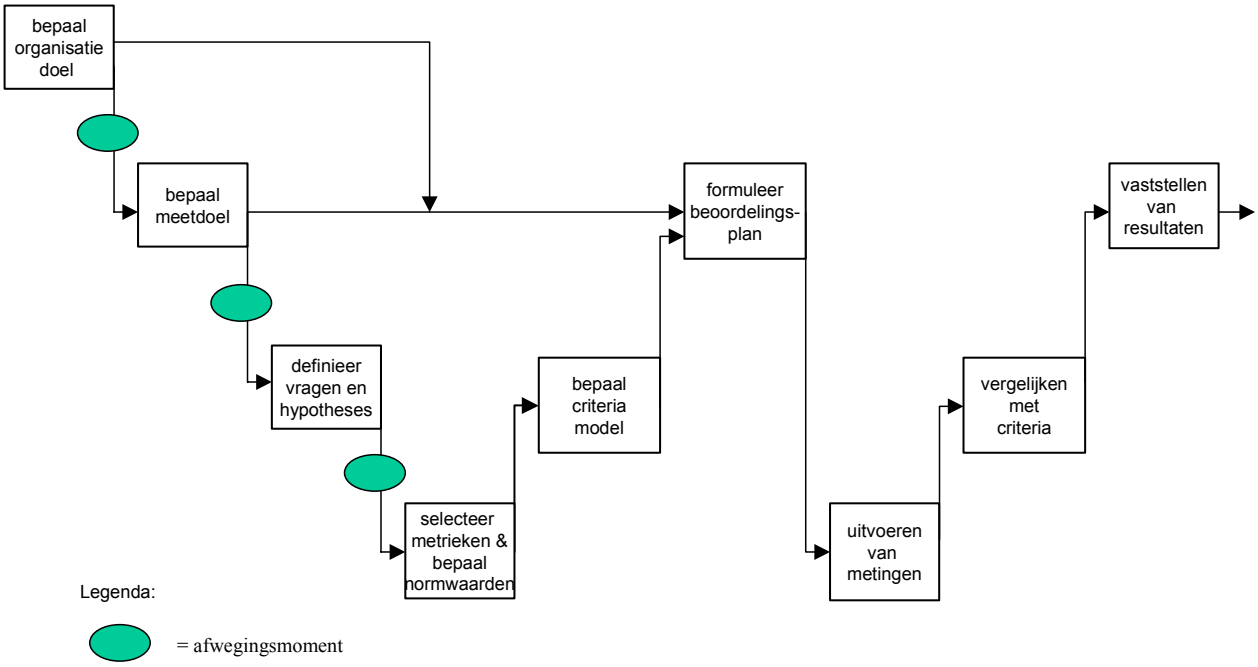
Na het ontwikkelen van de database is deze gevuld. Dit betrof het analyseren van bestaande referenties (Scope, 1992b), (ISO CD 9126 deel 2 en 3, 1999). Vervolgens zijn deze conform het datamodel gestructureerd en opgenomen in de database. De focus lag hierbij op één element van de ISO-kwaliteitskarakteristiek, namelijk: Maintainability’: onderhoudbaarheid. Softwareproduct-beoordelingen vergen namelijk veel domeinkennis, er is daarom gekozen voor het verzamelen van kennis op een beperkt gebied (Punter, 1998a), (Wassink, 1998). Ervaringen met het toepassen van de database komen in hoofdstuk 7 aan de orde.

Afstemmen van de benodigde en beschikbare middelen

De derde stap in de afweging betreft de feitelijke afstemming van de benodigde en beschikbare middelen. Ondanks de focus die in het voorgaande op metrieken en daarmee op de activiteit ‘selecteer metrieken en bepaal normwaarden’ ligt, moet er in principe gedurende het gehele proces worden afgewogen. Tijdens elke activiteit worden immers middelen ingezet. Het begrip afwegingsmoment is hierbij van belang. Dit is een moment in het beoordelingsproces waarop benodigde en beschikbare middelen worden afgestemd én op

basis waarvan vervolgacties worden gedefinieerd. Tijdens zo'n afwegingsmoment kan blijken dat de opgestelde vragen het actuele meetdoel slechts deels afdekken. Er moet dan vervolgens actie worden ondernomen. Een afwegingsmoment kan dus een review van de resultaten van een activiteit zijn. Het kan ook de inschatting met betrekking tot de uitvoerbaarheid van een activiteit zijn die vooraf aan het uitvoeren van deze activiteit wordt gemaakt.

In principe zijn de afwegingsmoment over het gehele proces te verdelen. Op basis van onze ervaringen tijdens de cases in hoofdstuk 4 en 5 de problemen stellen we echter dat de drie momenten om af te wegen zich veelal zullen concentreren op de eerste vier activiteiten uit de processtructuur. Dit is mede te verklaren uit het feit dat met name tijdens deze activiteiten de basis voor de inrichting van de beoordeling wordt gelegd en dat hierbij alle drie doel-middel-niveaus en daarmee de afwegingen aan de orde komen. De drie afwegingsmomenten die op de eerste vier activiteiten betrekking hebben worden in de onderstaande figuur uitgebeeld.



Figuur 6.7 Belangrijkste afwegingsmomenten in het proces

6.5 Bijsturing door het identificeren van terugkoppelingsinformatie

In deze paragraaf gaan we in op de problematiek die we hebben aangeduid als onvoldoende terugkoppeling en bijsturing.

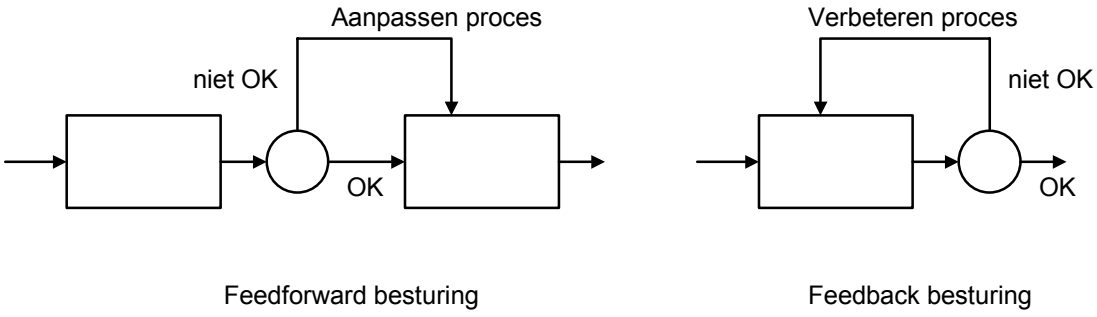
Besturingsmechanismen

Tot nu toe heeft het analysekader zich wat betreft de bijsturing gericht op de vraag óf er terug gekoppeld werd. In de cases hebben we echter geconstateerd dat er sprake is van verschillende soorten terugkoppeling. Zo kwam in de Omega-casestudie aan de orde dat er

tijdens reviews naar de procesresultaten wordt gekeken, terwijl ook het bijstellen van normwaarden (tuning) een vorm van terugkoppeling is. Met betrekking tot de reviews is het echter de vraag of het proces hiermee wordt bijgestuurd. Deze vraag maakt het noodzakelijk om de mechanismen van terugkoppeling en bijsturing eerst verder te analyseren. Daarvoor beschrijven we twee basismechanismen om een proces bij te sturen. Deze mechanismen zijn: feedforward en feedback besturing. Deze mechanismen stammen uit de besturingstheorie (de Leeuw, 1980), (In 't Veld, 1988). Beide worden gerekend tot de zogenaamde '*closed loop' of 'gesloten lus'* besturing.

Feedforward is voorwaartse koppeling. Men meet relevante invloeden op het proces in uitvoering waarna wordt geprobeerd op basis van deze kennis te voorspellen wat er mis kan gaan tijdens een volgende activiteit. Als er wordt verwacht dat er fouten gaan optreden, dan worden de volgende activiteiten aangepast. Feedforward vereist een procesmodel dat volledig is en als het ware alle invloeden in beeld heeft met de daarbij behorende repercussies. Als zodanig kan men precies oorzaak en gevolg overzien en op die basis het proces voorwaarts bijsturen. In de praktijk treft men maar zelden dergelijke ideaalsituaties aan en zal men veel meer gedwongen zijn om in plaats van feedforward te kiezen voor het principe feedback.

Feedback is terugwaartse koppeling. Het uitgevoerde proces wordt achteraf gecorrigeerd. Hiertoe bepaalt men of het proces tot het gewenste effect leidt en als zodanig het beoogde doel bereikt. Zolang dit niet het geval is wordt het proces verbeterd. De twee mechanismen van de 'closed loop' besturing –feedforward en feedback– worden in de onderstaande figuur naast elkaar gezet.

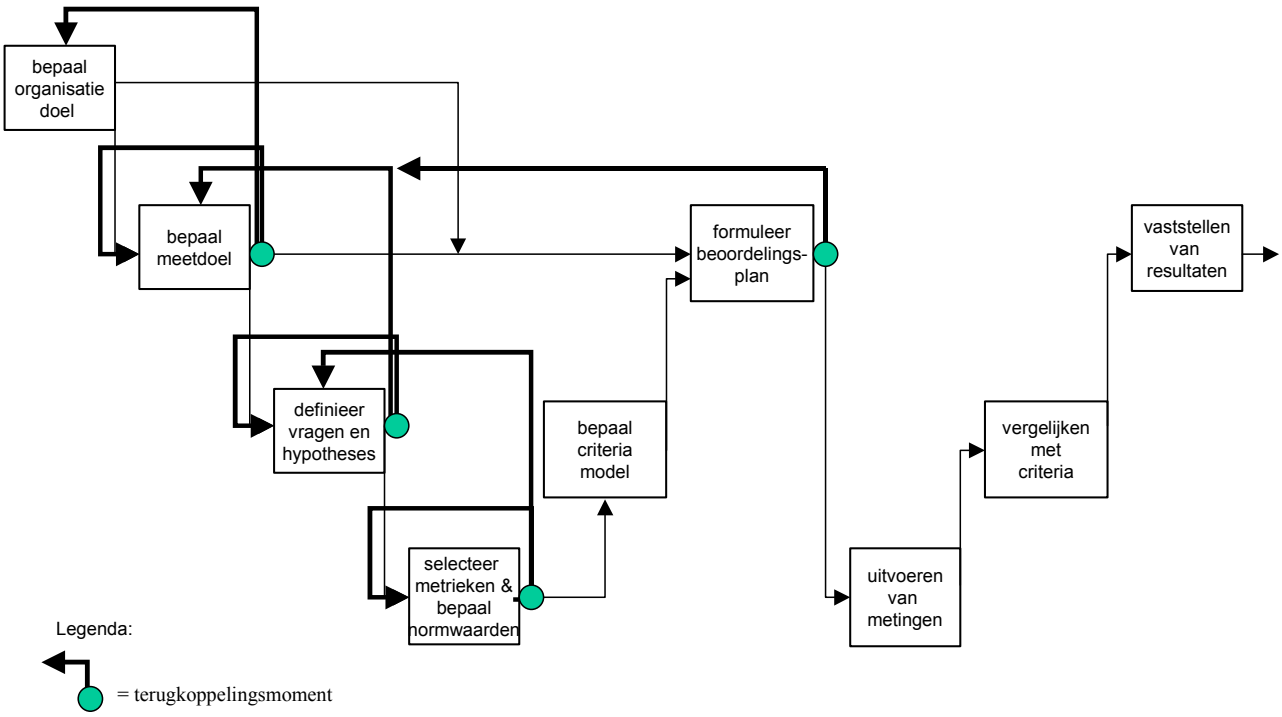


Figuur 6.8 Twee mechanismen om bij te sturen

Ten aanzien van beide 'closed loop'-mechanismen merken we nogmaals op dat het feedforward-mechanisme een model vereist waarin precies en volledig wordt beschreven welke invloeden er kunnen optreden en hoe kan worden ingegrepen in het proces. In de hoofdstukken hiervoor is geconstateerd dat onze kennis over modellen voor beoordelingsactiviteiten bij lange na niet voldoet aan deze eis en in die zin beperkt is. Voor de bijsturing van het beoordelingsproces moet daarom in principe het feedback-mechanisme

worden toegepast. Bij hoge uitzondering doen zich situaties voor dat men feedforward kan bijsturen en als zodanig preventief in plaats van correctief kan ingrijpen.

In principe is het mogelijk om vanuit elke activiteit in de processtructuur terug te koppelen op voorgaande, eerder uitgevoerde activiteiten. Op basis van de casestudies komen we echter tot een inperking. We zagen immers dat de meeste problemen zich voordoen in het begin van het proces. Dit leidde tot problemen in de rest van het proces. Terugkoppeling dient zich dan ook vooral te richten op de eerste activiteiten in het beoordelingsproces. Tijdens deze activiteiten wordt de basis van de beoordeling gelegd. Dit wordt uiteindelijk vastgelegd in het beoordelingsplan. Dit betekent dat er ook na het opstellen van dit plan dient worden teruggekoppeld. We geven deze meest noodzakelijke terugkoppelmomenten weer in de volgende figuur.



Figuur 6.9 Belangrijkste terugkoppelingsmomenten in de processtructuur

Maturity index on Reliability (MIR)

De wijze waarop feedback in een beoordelingsproces plaats vindt kan getypeerd worden met het Maturity Index on Reliability-model (MIR) dat door Brombacher is ontwikkeld (Brombacher, 1999), (Brombacher, 2000). Het MIR-model geeft vijf 'capability' of volwassenheidsniveaus waarmee de kwaliteit van de bijsturing wordt uitgedrukt, te weten:

- Niveau 0: ongecontroleerd – er zijn geen kwantitatieve gegevens over het proces. Aan dit eerste niveau voldoet ieder proces.
- Niveau 1: gemeten – er zijn kwantitatieve gegevens over het proces. Dat er problemen zijn is bekend, maar de oorsprong van deze problemen kent men niet.

- Niveau 2: geanalyseerd – de kwantitatieve gegevens over de procesresultaten en ook de oorsprong van de problemen en afwijkingen zijn nu bekend, maar helaas ontbreekt het nog aan inzicht over de oorzaken.
- Niveau 3: gecontroleerd –de oorsprong van de problemen worden onderkend, maar dezelfde afwijkingen worden bij het herhalen van het proces niet voorkomen.
- Niveau 4: continu verbeteren – het proces is in staat om te anticiperen op fouten uit het verleden en deze niet te herhalen tijdens het opnieuw uitvoeren van een proces.

Dat het MIR-model spreekt over volwassenheidsniveaus en dat dit er nota bene vijf zijn, maakt dat de vergelijking met het Capability Maturity Model (CMM) (Humphrey, 1989), (Paulk, 1995) snel wordt gelegd. Het CMM-model richt zich echter op het inventariseren van de vaardigheden van de organisatie (capabilities) om tot een meer volwassen software proces te komen. Het MIR-model richt zich op het analyseren van de informatiestromen voor de besturing van processen in organisaties.

Als we de MIR-niveaus toepassen op de beoordelingen die zijn gepresenteerd in de cases van hoofdstuk 4 en 5, dan constateren we dat hierbij vooral sprake was van terugkoppeling en bijsturing op MIR-niveau 1. Zo werden in beide beoordelingen de resultaten van een aantal activiteiten gereviewed. Hierbij ging het voornamelijk om het afwijzen of accepteren van de resultaten. Er vonden geen metingen plaats, ook werd niet naar de oorzaken en oorsprong van de problemen gezocht. Reviews in deze zin zijn weliswaar nuttig maar zijn onvoldoende om te kunnen spreken van terugkoppeling en bijsturing. Bij dergelijke reviews gaat het immers om een kritische reflectie of de resultaten wel of niet bevredigend zijn en als zodanig door betrokkenen worden geaccepteerd. Hoogstens kan er bij een onbevredigend resultaat worden besloten tot het opnieuw starten van het beoordelingsproces. Wil er echt sprake zijn van terugkoppeling en bijsturing dan moet een proces op zijn minst gesitueerd zijn op MIR-niveau 3. We lichten dat in het navolgende toe.

Operationalisering vanaf MIR-niveau 3 in beoordelingsprocessen
Feedback op niveau 3 veronderstelt drie activiteiten, namelijk: metingen, vaststellen probleem en vaststellen oorzaken.

Meten – meten aan een beoordelingsproces kan op verschillende manieren plaatsvinden. Het varieert van de simpele constatering dat er problemen in het proces zijn tot het uitvoeren van kwantitatieve metingen. Voor het verzamelen van algemene informatie over het beoordelingsproces kan gebruik worden gemaakt van de volgende metrieken:

- De kosten van de beoordeling – de kosten (in euro's, guldens e.d.) die gemoeid zijn met het uitvoeren van een beoordeling.
- Doorlooptijd van de beoordeling – het aantal dagen, weken of maanden waarin een beoordeling wordt uitgevoerd.

– Score van de tevredenheid over de beoordeling – deze score betreft de beantwoording van de vraag aan betrokken partijen of de beoordeling voldoet aan hun verwachtingen.

Om op niveau 3 problemen vast te stellen en oorzaken te kunnen duiden, dient ook op activiteiten-niveau en wellicht daarbinnen te worden gemeten.

Probleem vaststellen – voor het vaststellen van problemen is het zaak om op basis van de meetinformatie die activiteiten te identificeren die problemen in het proces veroorzaken. De processtructuur, zie figuur 6.4, kan hierbij dienst doen als referentie. Hiervoor is al aangegeven dat dit vooral de eerste vier activiteiten van het beoordelingsproces betreft.

Vaststellen oorzaken – om op basis van de meetinformatie de oorzaken van problemen vast te stellen, kan gebruik worden gemaakt van een overzicht met meest voorkomende root-causes (Brombacher, 1999). De onderstaande tabel geeft een aanzet van probleemoorzaken bij beoordelingsprocessen.

Tabel 6.1 Voorbeelden van probleemoorzaken per beoordelingsactiviteit

Activiteit	Probleemoorzaken
Bepaal organisatiedoel	Partijen zijn niet onderkend,
Bepaal meetdoel	Meetdoelen niet volgens template geformuleerd, meetdoelen niet geprioriteerd,
Definieer vragen en hypotheses	Vragen zijn te concreet (als metriek) of te abstract (als doel) geformuleerd,
Selecteer metrieken en bepaal normwaarden	Gebruik van metrieken zonder meetmodel,

Het voorgaande operationaliseert hoe feedback tenminste op niveau 3 gerealiseerd kan worden. Om eventueel niveau 4 (continu verbeteren) te bereiken is informatie over eerder uitgevoerde processen nodig. Deze informatie kan in metamodellen over het proces zijn opgeslagen. Met het opbouwen van dergelijke informatie wordt er geleerd over het proces.

Deze bijsturingsinformatie is bedoeld om beter bij te kunnen sturen. Dit impliceert dan wel dat er voldoende ingreepmogelijkheden bestaan om bij te kunnen sturen. Dit raakt aan het in hoofdstuk 3 behandelde begrip besturingsvariëteit.

Richtlijnen ter verbetering van de bijsturing

Op basis van het voorgaande komen we tot de volgende richtlijnen voor het verbeteren van de bijsturing:

- Onderken expliciet terugkoppelmomenten in het beoordelingsproces.
- Meet aan het proces, bepaal de problemen in het proces en stel de oorzaken van deze problemen vast om bijsturingsinformatie van voldoende volwassenheidsniveau te laten zijn.

6.6 Omgaan met beperkte rationaliteit en politiek

In het voorgaande hebben we de belangrijkste problemen besproken bij het beoordelen van software en is een groot aantal aanbevelingen gedaan om dat proces te verbeteren. Aan het eind van dit hoofdstuk willen we opnieuw reflecteren op aard en wezen van een beoordelingsproces maar nu meer vanuit het perspectief van beperkte rationaliteit en politieke besluitvorming.

Besturing van een beoordeling van een softwareproduct is te zien als een beslissingsproces waarbij wordt gekozen uit alternatieven met het oog op te bereiken doelen. In de ideale situatie weet de beslisser exact wat hij wil, heeft de beslisser alle informatie om beslissingen te nemen, wordt hij geacht alle mogelijke alternatieven te overzien en is zijn vermogen tot informatieverwerking onbegrensd (De Leeuw, 1980). In de beoordelingspraktijk is hiervan echter geen sprake. In plaats daarvan worden beslissingen genomen met een *beperkte rationaliteit*. De beslisser weet niet zo precies wat hij wil: de doelstellingen zijn wazig of onbekend. De beslisser heeft ook geen of niet genoeg informatie om beslissingen te nemen en hij overziet niet alle mogelijke alternatieven en zijn vermogen tot informatieverwerking is begrensd. Besturing met een beperkte rationaliteit duiden we ook aan als ‘intuïtieve’ of ‘primitieve’ besturing (Heemstra, 1989). Dergelijke besturing vindt zijn oorsprong in de onduidelijke of ‘fuzzy’-omgeving van de beoordelingen. We doelen hier vooral op de impliciete relaties tussen externe en interne metrieken (zie hoofdstuk 2) en de contextafhankelijkheid wat nu precies goede software-kwaliteit is.

Naast het niet-rationele karakter van de besturing van beoordelingsprocessen is er ook sprake van een politiek karakter. In de casestudies is dit aan de orde gekomen bij het formuleren van doelen. Er bleek toen dat er verschillende actoren zijn die elk hun eigen belangen trachten te verwezenlijken.

Het beperkt rationele en het politieke karakter van een beoordelingsproces leiden tot aanvullingen op het inmiddels opgestelde ontwerp uit de voorgaande paragrafen.

Beslissen in een beperkt rationeel en politiek proces

Ook al heeft de besturing van een beoordelingsproces niet een puur rationeel karakter, toch zullen er beslissingen moeten worden genomen. Renkema (1996) schetst vier onderwerpen die daarbij betrokken kunnen worden, nl:

- Product – de inhoudelijke argumenten van de beslissingen.
- Proces – de fasering of volgorde van de stappen waarin de beslissing wordt genomen.
- Participatie – het betrekken van de juiste (groepen van) personen bij het nemen van de beslissingen.
- Politiek – het erkennen dat er belangentegenstellingen mogelijk zijn.

Hieronder gaan we in op de relevantie van elk van deze onderwerpen voor de besturing van een beoordelingsproces.

Product – dit betreft de inhoudelijke argumenten op basis waarvan beslissingen genomen worden. Hiervoor is al aangegeven dat de fuzzy omgeving waarin beoordelingen veelal worden uitgevoerd, het moeilijk maakt om op dit gebied in te grijpen. Het ontbreekt aan juiste informatie en aan goede modellen. Dit kwam bijvoorbeeld tot uiting in de Omega-casestudie waar de metrieken uit een bestaande referentie worden overgenomen om op basis van het vertrouwen van de engineers in het betreffende model. Waarop dat vertrouwen was gebaseerd, werd niet geëxpliciteerd.

Proces – dit onderwerp gaat over de weg waarlangs beslissingen tot stand komen (procedurele rationaliteit). 'Moet men daartoe eerst het doel bepalen, vervolgens informatie verzamelen en dan alternatieven ontwikkelen?' (De Leeuw, 1980). In heel wat gevallen verloopt besluitvorming niet zo rationeel en zal men bijvoorbeeld alternatieven ontwikkelen en daaruit keuzes maken voordat sprake is van een duidelijk omschreven doel. Achteraf worden er dan doelen 'bij verzonnen' om alsnog de genomen beslissingen te rationaliseren.

Participatie – dit betreft de structurele rationaliteit. Er is veelal niet sprake van één enkele beslisser. De beslissing is het resultaat van een proces waarin meerdere beslissers participeren. Deze beslissers –i.c. de partijen– dienen tevreden te zijn over de te nemen beslissingen. Het is dan ook zaak om de samenwerking tussen beslissers zo te organiseren dat er voldoende draagvlak is voor het accepteren van de beslissingen.

Politiek – dit is een onderwerp dat door Renkema wordt toegevoegd aan de drie voorgaande onderwerpen Renkema geeft aan dat verschillende actoren verschillende eigen wensen en voorkeuren hebben. De verdeling van lasten en baten van een (infrastructuur-investering) beslissing beïnvloeden sterk de politieke geladenheid ervan. In dit verband spreekt Renkema over winst- en verliespunten van een beslissing.

Uitwerking

In het voorgaande zijn vier gebieden bepaald waarop kan worden ingegrepen op de besluitvorming tijdens een beoordelingsproces, namelijk: product, proces, participatie en politiek. Welke mogelijke ingrepen toegepast kunnen worden, wordt in het navolgende kort aangegeven.

Product – zoals eerder aangegeven is het verstandig een besluit zo veel als mogelijk met rationele argumenten te onderbouwen en die argumenten open te communiceren naar betrokken partijen. Daarmee onstaat geen rationele besluitvorming, maar tracht men het ideaal van een objectieve en rationele besluitvorming zo veel als mogelijk te benaderen.

Proces – beslissingen in een beoordelingsproces vinden plaats op verschillende tijdstippen. Daarvoor is het niet strikt noodzakelijk om een proces conform de eerder beschreven aktiviteiten te doorlopen. Er bestaan met andere woorden verschillende uitvoeringsstrategieën. Ook voor software- en informatiesysteemontwikkeling worden verschillende strategieën beschreven (Boehm, 1988), (Bersoff en Davis, 1991), (Lemmen, e.a., 1993). Uit deze referenties leiden we drie strategieën af voor het nemen van beslissingen in een beoordelingsproces, namelijk:

- Lineair iteratieve strategie – het beoordelingsproces wordt gezien als een éénmalige rondgang door de processtructuur. Men begint bij de eerste activiteit en loopt vervolgens de verschillende activiteiten lineair, achter elkaar, door. Na afloop van elke activiteit wordt er teruggekoppeld.
- Inherent iteratieve strategie – in deze strategie kan er vanuit elke activiteit in het beoordelingsproces worden gestart naar een volgende stap. Het maakt in principe niet uit met welke activiteit men begint. Na een aantal iteratieslagen moet echter voor elke activiteit beslist zijn hoe de activiteit in te vullen. De in- en uitvoerstromen van de verschillende activiteiten moeten op elkaar aansluiten zodat de keten van activiteiten gesloten blijft.
- Evolutionaire strategie – in deze strategie wordt de beoordeling via verschillende versies uitgewerkt. Middels verschillende versies wordt de beoordeling uitgebouwd tot een beoordeling waarover men tevreden is. Er is een eerste beoordeling maar die is nog lang niet compleet en definitief. De beoordeling evolueert voortdurend, waarbij de processtructuur meerdere keren wordt doorlopen.

Elk van deze drie strategieën heeft gevolgen voor het nemen van beslissingen. Zo wordt er in de lineair iteratieve strategie in principe niet teruggekomen op een eenmaal genomen beslissing. In de inherent iteratieve en zeker de evolutionaire strategie is het juist wel mogelijk om terug te komen op eenmaal genomen beslissingen. Er dient dan ook vooraf aan het doorlopen van de processtructuur bepaald te worden welke strategie gevolgd wordt. De keuze van de strategie hangt ons inziens af van de onzekerheid bij de mensen over wat er tijdens het proces gedaan moet worden (wat is het doel), en hoe dit bereikt moet worden. Als de onzekerheid over doel en proces groot is dan is een evolutionaire strategie aan te bevelen, omdat er immers verschillende versies van de beoordeling ontwikkeld kunnen worden. De keuze voor deze strategie zal echter ook afhangen van de beschikbare tijd en middelen om op beslissingen terug te komen. Als men het doel van de beoordeling in één keer kan formuleren en als de wensen van de opdrachtgever duidelijk zijn, dan ligt het voor de hand om de lineaire strategie te volgen.

Participatie – om samenwerking te organiseren is het zaak om maatregelen te treffen om de juiste mensen op het juiste moment bij het beoordelingsproces te betrekken. Om de mate van participatie in een proces vast te stellen en maatregelen ter verhoging van participatie te treffen is inzicht in de situationeel bepalende factoren nodig. Interessante bron in deze is de

literatuur rondom contingentie- en situationele factoren voor projecten (Haas en Wubbels, 1990), (Verhoef en Van Swede, 1995), (Van Swede en Van Vliet, 1992) en kritieke succesfactoren van meetprogramma's (Hall en Fenton, 1997), (Niessink, 2000). Op basis van de vele factoren in deze literatuur, komen we tot vier belangrijke factoren die volgens ons de participatie tijdens een beoordelingsproces beïnvloeden, namelijk:

- Motivatie – hieronder verstaan we de bereidheid van personen of groepen mensen om zich voor een beoordeling in te zetten. Motivatie is een complex begrip (Markus, 1984) dat op zijn beurt door vele andere factoren beïnvloed wordt. Wij veronderstellen dat er van een groep mensen ingeschat kan worden wat hun motivatie is cq. hoe deze motivatie positief beinvloed kan worden.
- (Management) commitment – dit betreft de financiële ruimte en de aandacht die het management van een organisatie toekent aan een beoordeling. Beide drukken het belang van het management uit en daarmee de mate waarin men serieus naar de beoordeling(sresultaten) zal kijken,
- Gebruik van tools – hierbij gaat het om de ondersteuning van mensen bij het uitvoeren van beoordelingstaken. Dit kan zowel de ontwikkeling als de uitvoering van de beoordeling betreffen. Een speciale groep tools betreft zogenaamde 'collaborative' tools, waarmee mensen met elkaar samen kunnen werken. Denk hierbij aan groupware in het algemeen en aan Group Decision Support Systemen in het bijzonder (Bongers en Geurts, 1998). Verwacht wordt dat dergelijke tools positief doorwerken op participatie. Daarnaast zal ook de aantrekkingskracht van dergelijke tools op betrokkenen een rol spelen, zeker bij beoordelingen waarin engineers of andere technisch georiënteerde personen zijn betrokken. Een gevleugelde uitspraak in dit verband is ´use tools to get practioners aboard´ (Deligiannis e.a, 1999).
- Aanbieden van referenties – het zelf van scratch af opzetten van een beoordeling vergt veel energie en tijd van mensen. Vandaar dat het zinnig is om bestaande kennis aan te bieden, zodat de betrokkenen niet het idee krijgen dat 'het wiel opnieuw wordt uitgevonden'.

Politiek – om winst- en verliespunten te bepalen, is het van belang dat eerst de partijen in een beoordeling bekend zijn. In paragraaf 6.2 is hier al aandacht aan besteed. Om partijen te inventariseren en te identificeren is een actieve rol van de beoordelaar vereist. De beoordelaar dient de rol van moderator aan te nemen die zich opwerpt als aanstuwer van de beoordeling. De rol van moderator in een beoordeling komt overeen met de moderator in Fagan-inspecties (Fagan, 1978). De beoordelaar stelt zich in deze dus niet op als 'politieagent' maar is meer een bemiddelaar.

Richtlijnen voor het omgaan met beperkte rationaliteit en het politieke karakter van beoordelingsproces

Op basis van het voorgaande komen we tot de volgende aanvullende richtlijnen voor het verbeteren van de bijsturing:

- Bepaal een strategie voor het doorlopen van het beoordelingsproces.
- Bepaal de mate van participatie van de betrokkenen bij het beoordelingsproces en tref maatregelen indien deze participatie te gering dreigt te worden.
- Inventariseer winst- en verliespunten voor de onderscheiden partijen en ga na of deze in een redelijke verhouding tot elkaar staan. Zo niet, zoek dan naar een andere verhouding en tracht meningsverschillen tussen partijen te beslechten.Hierbij is meestal een belangrijke rol weggelegd voor de beoordelaar in zijn rol als moderator.

6.7 Het ontwerp samengevat

In dit hoofdstuk is een ontwerp voor doelgericht beoordelen opgesteld. Dit ontwerp is gericht op het ondervangen van de vier problemen ten aanzien van de besturing van beoordelingsprocessen, die in hoofdstuk 3 tot en met 5 aan de orde zijn gekomen.

De basis van het ontwerp vormt de processtructuur die in paragraaf 6.3 is beschreven. Deze structuur geeft de uit te voeren beoordelingsactiviteiten en de relaties daartussen. De processtructuur alleen is onvoldoende voor een doelgerichte beoordeling. Er zijn daarom richtlijnen uitgewerkt, deze zijn:

1. Identificeer partijen aan de hand van standaardrollen in een proces.
2. Operationaliseer het beoordelingsdoel door het als meetdoel met behulp van het measurement goal template te formuleren.
3. Voer versiebeheer van doelen uit.
4. Bepaal winst- en verliespunten in een beoordeling om het verwachtingspatroon van de partijen te managen; de beoordelaar is hierbij moderator.
5. Bepaal de strategie voor het doorlopen van de activiteiten in het beoordelingsproces.
6. Zorg ervoor dat de voor een beoordeling geselecteerde mensen ook daadwerkelijk aan het beoordelingsproces deelnemen (participeren).
7. Onderken expliciet afwegingsmomenten in het proces: de momenten waarop in het proces doel en middelen worden afgewogen.
8. Expliciteer doel en middelen. Het doel wordt geëxpliciteerd door het te formuleren middels het measurement goal template. De middelen worden geëxpliciteerd door ze te karakteriseren met behulp van attributen. Voorbeelden zijn het object of de kwaliteitskarakteristiek waarop het middel betrekking heeft.
9. Onderken expliciet terugkoppelmomenten in het proces.
10. Meet aan het proces, bepaal de problemen in het proces en stel de oorzaken van deze problemen vast om bijsturingsinformatie van voldoende volwasenheidsniveau te laten zijn.

Hieronder geven we een overzicht van procesactiviteiten in samenhang met de richtlijnen. We onderkennen hierbij de vier oorspronkelijke probleemgebieden.

Doelformulering – hierbij gaat het allereerst om de activiteiten ‘bepalen organisatiedoel’ en ‘bepalen meetdoel’. Bij de laatste activiteit hebben we te maken met de richtlijn ‘gebruik measurement goal template’. Voor beide activiteiten zijn er richtlijnen:

- Onderkennen van partijen aan de hand van standaardrollen.
- Versiebeheer van doelen.
- Bepalen van winst en verliespunten. De laatste richtlijn komt voort uit paragraaf 6.6: omgaan met beperkte rationaliteit en met het politieke karakter van beoordelingen.

Strategie en aansturing – de verbetering van de aansturing is gezocht in het expliciet onderkennen van doel-middel relaties in een beoordelingsproces. Vervolgens is een nieuwe processtructuur opgesteld. Het is zaak deze structuur in acht te nemen bij het inrichten van een beoordeling. Voorafgaand aan het inrichten moet een strategie voor het doorlopen van het proces worden bepaald. Tijdens het doorlopen van het proces is het belangrijk om de juiste mensen op het juiste tijdstip erbij te betrekken. In paragraaf 6.6 zijn vier factoren onderkend die van invloed zijn op de participatie van mensen. Strategie en participatie betreffen richtlijnen die voortkomen uit paragraaf 6.6: omgaan met beperkte rationaliteit en met het politieke karakter van beoordelingen

Afweging – in de processtructuur zijn diverse afwegingsmomenten onderkend. Dit zijn momenten in het proces waarop er zeker moet worden nagedacht over de inzet van middelen in relatie tot het gestelde doel. Voor de afweging zelf is het zaak om zowel doel als middelen te expliciteren. Dit betekent dat we het doel nader omschrijven met behulp van het goal measurement template. Aan de middelen kennen we attributen toe. In paragraaf 6.4 is een een set attributen voorgesteld voor een bepaald type middelen, namelijk metrieken. Er zijn vijf attributen voor metrieken onderkend: object, kwaliteitskarakteristiek, perspectief, reproduceerbaarheid en validiteit. Tenslotte is voorgesteld om informatie over metrieken bij te houden in een metriekendatabase, zodat dergelijke metrieken beter op het doel kunnen worden afgestemd.

Bijsturing en terugkoppeling – in de door ons ontworpen processtructuur zijn vier terugkoppelingsmomenten gedefinieerd. Op deze momenten moet vrijwel zeker worden teruggekoppeld. Vervolgens zijn er vijf volwasenheidsniveaus aan de bijsturing onderkend. Om met terugkoppelingsinformatie te sturen moet het proces aan de hoogste twee volwassenheidsniveaus voldoen. Dit betekent dat er wordt gemeten, dat problemen in het proces worden vastgesteld en dat de oorzaken worden bepaald. Door ervaringskennis van beoordelingsprocessen vast te leggen, wordt geleerd van voorgaande keren en wordt herbruikbare kennis gegenereerd voor toekomstige beoordelingsprocessen.

7. Evaluatie van het ontwerp om doelgericht te beoordelen

7.1 Inleiding en verantwoording

In het vorige hoofdstuk is een ontwerp voor doelgericht beoordelen opgesteld. Het ontwerp omvat de processtructuur en een set richtlijnen. In dit hoofdstuk willen we aannemelijk maken dat het ontwerp voldoende perspectief biedt als aanpak voor de in de hoofdstukken 3 tot en met 5 geschetste problematiek rondom de besturing van beoordelingsprocessen. Hiervoor onderkennen we drie niveaus waarop we het ontwerp evalueren, namelijk:

- Is er sprake van een logisch afgeleid ontwerp?
- Is het ontwerp toepasbaar?
- Zijn betrokkenen tevreden over het ontwerp?

Logisch afgeleid – dat er sprake is van een logisch afgeleid ontwerp moet duidelijk worden door expliciete relaties te leggen tussen de problematiek die aanleiding voor het onderzoek was en de oplossing die hiervoor is opgesteld. In hoofdstuk 6 is met het formuleren van de processtructuur en de richtlijnen ingegaan op elk van de vier besturingsproblemen. Hiermee is aannemelijk gemaakt dat met het ontwerp elk van de vier problemen wordt aangepakt.

Toepasbaarheid – dat een ontwerp toepasbaar is moet duidelijk worden door het ontwerp in de praktijk te implementeren en na te gaan of het gebruikt kan worden om zo te constateren of er wijzigingen of aanvullingen nodig zijn. De processtructuur en twee van de in hoofdstuk 6 geformuleerde richtlijnen zijn actief toegepast in de praktijk. Het betrof:

- Operationaliseren van doelen en het gebruik van het measurement goal template hierbij.
- Het expliciteren van middelen tijdens het afwegingsproces, en dan gericht op metrieken, door de metriekendatabase te gebruiken.

De overige acht richtlijnen zijn niet actief toegepast. Toch kunnen we ook over de toepasbaarheid van deze richtlijnen iets zeggen door te kijken of acties die uit deze richtlijnen voortvloeien überhaupt in de beschouwde beoordelingssituaties voorkomen. Een voorbeeld hiervan is de richtlijn om participatie te bevorderen. In hoofdstuk 6 zijn vier factoren geformuleerd die participatie beïnvloeden. Bij de toepasbaarheid van deze richtlijn in dit hoofdstuk laten we zien dat ze ook werkelijk een rol spelen. Daarmee wordt de betreffende richtlijn aannemelijk gemaakt. Andere richtlijnen die niet actief zijn toegepast maar die toch aan de orde komen zijn:

- Identificeer betrokken partijen aan de hand van standaardrollen in het proces.
- Voer versiebeheer van doelen uit.
- Bepaal een strategie voor het doorlopen van het proces.

- Onderken expliciet afwegingsmomenten in het proces.
- Onderken expliciet terugkoppelmomenten in het proces.
- Meet aan het proces, bepaal de problemen in het proces en stel de oorzaken van deze problemen vast om bijsturingsinformatie van voldoende volwasenheidsniveau te laten zijn.

De ervaringen met deze richtlijnen en de processtructuur worden gepresenteerd door per ontwerpaspect -doelformulering, strategie en aansturing, afweging van doel en middelen en terugkoppeling- in te gaan op de verschillende opdrachten.

Tevredenheid – voor richtlijnen die actief zijn toegepast is nagegaan of betrokken personen daarover tevreden zijn. De tevredenheid is een belangrijke voorwaarde voor acceptatie (face validity) van het ontwerp in de praktijk. De tevredenheid is vastgesteld door mensen te bevragen.

Richtlijnen uit het ontwerp zijn toegepast tijdens opdrachten bij drie verschillende organisaties, namelijk: Océ Technologies, Tokheim RPS en Cap Gemini ISM. Bij de eerste twee organisaties betrof de opdracht het opstellen van een nieuw metriekenprogramma om de kwaliteit van (embedded) code te beoordelen. Bij Cap Gemini betrof de opdracht het formuleren van een nieuwe checklist voor een bestaande audit om de Maintainability van informatiesystemen te bepalen.

Code metrics beoordeling bij Océ
Deze opdracht betrof de ontwikkeling van een beoordeling van software in copiers. De opdracht is uitgevoerd bij Océ Nederland, afdeling Technologies, Research and Development. Océ ontwikkelt kopieerapparaten (copiers), printers en scanners. Het bedrijf richt zich op verschillende segmenten van de markt voor kopieerapparaten en heeft een leidende positie op het gebied van het maken van grote afdrukken. De organisatie wil deze positie behouden en proberen uit te breiden.

Omdat software een steeds belangrijker onderdeel van copiers wordt, werd door het management van Océ vastgesteld dat de kwaliteit ervan beter beheerst en verbeterd moest worden. Daarom werd een aantal projecten gedefinieerd, waarvan het project ‘statisch testen van code’ er één was. Het doel van dit project was om te bekijken of statisch testen een effectieve maatregel is voor het verbeteren van de software-kwaliteit. Binnen Océ werd ad hoc al gebruik gemaakt van codemetrieken en van het tool QAC (Programming Research, 1994). Met het project ‘statisch testen’ werd overwogen om dit uit breiden. Hierbij stond de vraag centraal of codemetrieken een goede voorspeller van softwareproductkwaliteit zijn. Onder softwareproductkwaliteit werd verstaan: Maintainability en Reliability.

Voor het beantwoorden van deze vraag is besloten om binnen Océ-Technologies twee software ontwikkeltrajecten aan te wijzen waar ervaring met het toepassen van codemetrieken moest worden opgedaan, namelijk het Boston- en het Dac 116-project.

Het Boston-project betrof de ontwikkeling van een nieuwe versie van een bestaande copier voor de grafische markt. Hiermee kunnen grote afmetingen papier gekopieerd worden. De software in deze copier omvatte 9 modules. Het Boston-team ontwikkelt alle elementen van de nieuwe versie van de copier.

Het Dac 116-project betrof de ontwikkeling van een netwerk voor copier/printers en scanners. De Dac 116 omvatte 180 software modules. Hiervan werd een deel –ongeveer 10%– extern ingekocht. Een groot deel van de modules (80%) is overgenomen uit een bestaande copier/printer.

De opdracht voor de code metrics beoordeling bij Océ is uitgevoerd van augustus tot november 1999. Hiervoor is een experiment uitgevoerd waarbij de resultaten van (vijf) metrieken zijn vergeleken met de meningen van de ontwikkelaars van de software én met ervaringsgegevens over het aantal (ontwikkel)uren per module (voor Maintainabiliy) en het aantal fouten per module (voor Reliability). De onderzoeker heeft checklists opgesteld om de opinie van de ontwikkelaars te meten. Ook zijn de resultaten van deze metingen mede door hem geïnterpreteerd.

Code metrics beoordeling bij Tokheim RPS
Deze opdracht betrof een advies voor een nieuwe set metrieken die de bestaande selectie bij Tokheim RPS aanvullen. De opdracht is uitgevoerd bij hetzelfde bedrijf en rondom hetzelfde product, beschreven in hoofdstuk 4. De opdracht lag dan ook in het verlengde van de eerder behandelde casestudie. De opdracht is uitgevoerd van september 1998 tot maart 1999. Het heeft geresulteerd in een nieuwe set metrieken om de kwaliteit van de Omega-modules te bepalen.

Maintainability audit bij Cap Gemini
Deze opdracht betrof het opstellen van een nieuwe checklist voor een bestaande beoordelingsmethode bij Cap Gemini. Deze Application Maintainability Audit (AMA) is een dienst die werd aangeboden door de divisie Information Systems Management van Cap Gemini. De audit werd door Cap Gemini verkocht als een dienst ‘om in korte tijd een gedegen en helder inzicht in de onderhoudbaarheid van applicaties te krijgen’. Hieronder werd de technische kwaliteit verstaan alsook de mate waarin het systeem beheersbaar blijft in de toekomst.

De Application Maintainability Audit is voortgekomen uit analyses die Cap Gemini in het verleden maakte als de organisatie het onderhoud van een bestaand informatiesysteem op

zich nam. Voordat dit onderhoud werd aanvaard, bepaalde men de onderhoudbaarheid en daarmee de in de toekomst benodigde inspanning. De Application Maintainability Audit is ontstaan uit dergelijke analyses maar is in de opzet bedoeld als audit voor het geven van onafhankelijk advies.

Bij de aanvang van de opdracht was de Application Maintainability Audit zeven keer uitgevoerd bij diverse klanten, waaronder bedrijven op het gebied van telecom, energie en transport. Ondanks het feit dat de audits naar tevredenheid werden uitgevoerd, voelde Cap Gemini anno 1999 de behoefte om de audit te verbeteren. Men dacht hierbij aan het beter structureren van de audit, het meer eenduidig stellen van vragen en het opstellen van 'hardere' normen.

De opdracht is uitgevoerd van maart tot juni 1999. Er zijn nieuwe checklists opgesteld. Deze checklists zijn toegepast door beoordelaar(s) bij Cap Gemini tijdens een beoordeling van een pensioeninformatiesysteem van een verzekeringsmaatschappij.

De rol van de onderzoeker
In alle drie de opdrachten werd de onderzoeker gevraagd om een advies te geven over de door de organisatie uit te voeren beoordeling. De onderzoeker bracht hierbij zijn kennis rondom beoordelingsprocessen en -middelen (met name checklists en metrieken) in. De hieruit resulterende beoordelingen zijn uiteindelijk door de organisaties zelf uitgevoerd.

7.2 Evaluatie van de toepasbaarheid van het ontwerp

7.2.1 Doelformulering

Rondom het aspect doelformulering komen drie richtlijnen aan de orde. De richtlijn 'operationaliseren van beoordelingsdoel, door het als meetdoel te formuleren en hierbij het measurement goal template te gebruiken' is doelbewust toegepast. De richtlijnen 'identificeer partijen aan de hand van standaardrollen in proces' en 'voer versiebeheer van doelen uit' zijn niet toegepast, maar ten aanzien hiervan werd geconstateerd dat acties die met de richtlijnen beoogd werden, in de beoordelingssituaties voorkwamen.

Operationaliseren van beoordelingsdoel, door het als meetdoel te formuleren
Bij aanvang van de opdracht bij Océ waren er al twee meetdoelen geformuleerd. Deze waren niet door de organisatie zelf geformuleerd, maar overgenomen uit een bestaand meetprogramma van een andere organisatie. De Océ-organisatie vond deze meetdoelen geschikt voor haar eigen situatie. Voor de andere organisaties zijn de meetdoelen wel geformuleerd tijdens het uitvoeren van opdrachten. Hiervoor zijn diverse personen in de organisatie geraadpleegd waarna de onderzoeker het doel heeft geformuleerd. Er is gebruik gemaakt van het measurement goal template, zoals beschreven in hoofdstuk 6. De

geformuleerde doelen zijn vervolgens aan de mensen in de organisatie voorgelegd om hun instemming te krijgen. Onderstaande tabel geeft de meetdoelen.

Tabel 7.1 Overzicht van de meetdoelen tijdens de drie opdrachten

	Meetdoelen bij Océ	Meetdoel Tokheim RPS	Meetdoel Cap Gemini
Analyseer (object)	Source code	Source code	Product (informatiesysteem) en proces
Met als doel	Bepalen van de bruikbaarheid van codemetrieken	Bepalen wanneer een module te accepteren of af te wijzen	Bepalen hoe gemakkelijk of moeilijk het is om –delen van– het systeem te wijzigen
Quality focus	Maintainability en Reliability	Maintainability	Maintainability
Vanuit het gezichtspunt	Ontwikkelaars	Ontwikkelaars	Opdrachtgever van audit
In the context of	Océ and Dac project	Omega project	Beoordeling van pensioensysteem bij verzekeringsmaatschappij

Identificieer partijen aan de hand van standaardrollen in proces

Voor de drie opdrachten zijn achteraf de vier standaardrollen van een beoordeling ingevuld. De resultaten worden in de onderstaande tabel gepresenteerd.

Tabel 7.2 Beoordelingsrollen tijdens de drie opdrachten

	Opdrachtgever beoordeling	Beoordelaar	Producent software	Afnemer software
Océ	Werkgroep ‘statisch testen’ binnen Océ	Ontwikkelaars / tool	Ontwikkelaars	Afnemer binnen Océ
Tokheim RPS	Groep initiatief-nemers tot toolbeoordeling	Ontwikkelaars / tool	Ontwikkelaars	Opco, afnemer binnen Tokheim RPS
Cap Gemini	Verzekerings-maatschappij	Cap Gemini	Externe software ontwikkelaar	Verzekerings-maatschappij

In elk van de drie opdrachten is er maar één partij die het doel van beoordeling vaststelt. Het gaat in alle gevallen om de opdrachtgever van de beoordeling. Het zijn de Werkgroep ‘statisch testen’ binnen Océ, de groep initiatiefnemers binnen RPS om een geautomatiseerde kwaliteitsbeoordeling op te zetten en de verzekeringsmaatschappij die opdracht tot de beoordeling geeft. In alle gevallen is sprake van een klant die een opdracht geeft aan de beoordelaars. Er is daarmee een duidelijke verhouding tussen beoordelaar en opdrachtgever. Andere partijen –zoals afnemers van de software– bemoeien zich niet met de formulering. Omdat er maar één partij is die de doelen definieert, wordt deze partij niet gehinderd door andere partijen. Het is dan ook niet noodzakelijk om het doel in meer algemene termen te

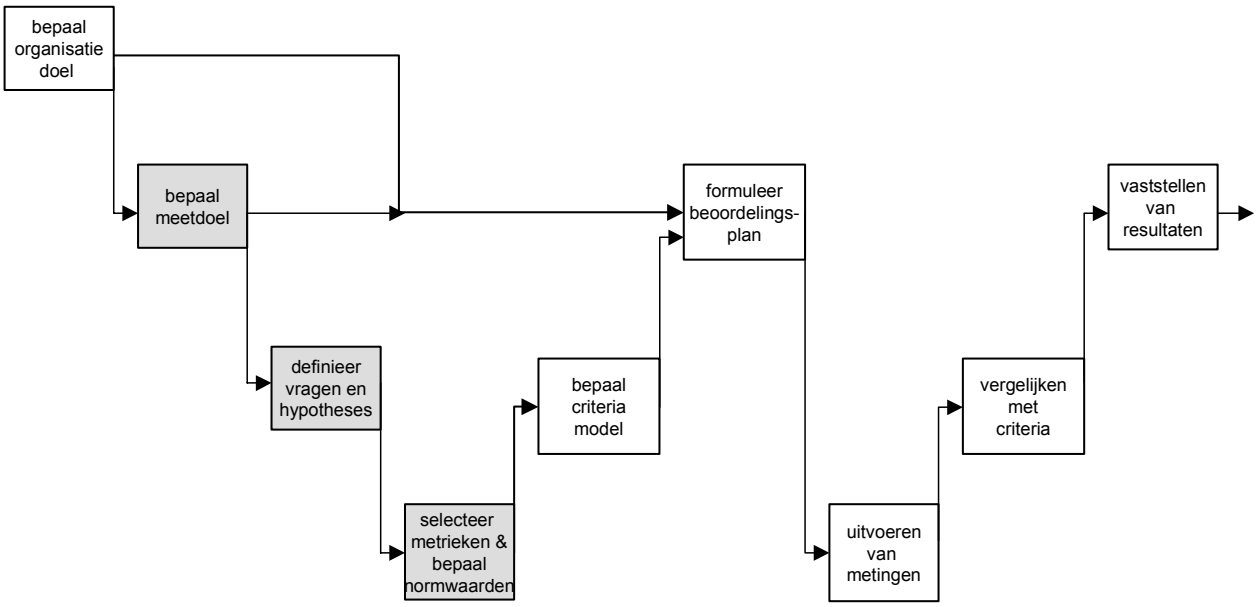
formuleren om daarover consensus te bereiken. De doelstelling kan concreet, als meetdoel, worden geformuleerd.

Voer versiebeheer van doelen uit
Gedurende de beoordelingsprocessen in de drie opdrachten kwamen de meetdoelen voortdurend aan de orde. Hierbij viel op dat van de vijf elementen waaruit een meetdoel bestaat het meest werd verwezen naar 'with respect'/'quality focus' en 'analyse'/'object'. Dit gebeurde door alle betrokken partijen. Zo werd het meetdoel tijdens de opdracht bij Océ door de ontwikkelaars steeds gevat in termen van Reliability en Maintainability (quality focus) en code (object). De andere drie elementen van het meetdoel (purpose, viewpoint en context) werden wel geformuleerd, maar werden verder niet ter discussie gesteld. Uit het voorgaande blijkt voor versiebeheer van doelen dat het bijhouden van versies van (meet)doelen in ieder geval de quality focus en het object dient te betreffen.

7.2.2 Strategie en aansturing

Bij het evalueren van de toepasbaarheid van het tweede aspect van het ontwerp richten we ons op de vraag welke activiteiten uit de processtructuur zijn uitgevoerd. Vervolgens gaan we in op de strategie. De richtlijn luidt: 'er dient voorafgaand aan het doorlopen van het proces een strategie te worden bepaald'. In ons onderzoek hebben we gekeken welke van deze strategieën gevolgd zijn tijdens de drie beoordelingen. Tenslotte komen de factoren die participatie beïnvloeden aan de orde. Dit wordt gedaan door vast te stellen of de vier, in hoofdstuk 6 onderkende, factoren een rol speelden tijdens de uitvoering van de drie beoordelingen.

Processtructuur
In de beschouwde beoordelingen werden niet steeds dezelfde activiteiten doorlopen. Zo werd er tijdens de opdracht bij Cap Gemini naast doelformulering, vraagdefinitie en metriekselectie ook een beoordelingsplan geformuleerd, werden de metingen uitgevoerd, werd er vergeleken met criteria en worden resultaten vastgesteld. Dit gebeurde niet tijdens de beoordeling bij Océ en Tokheim RPS. De drie beoordelingen besteden alle wel aandacht aan de activiteiten 'bepaal meetdoel', 'definieer vragen en hypotheses' en 'selecteer metrieken en bepaal normwaarden'. Dit wordt in de volgende figuur uitgebeeld.



Figuur 7.1 De nadruk van het proces tijdens de opdrachten ligt op drie activiteiten.

De activiteiten ‘definieer vragen en hypotheses’ en ‘selecteer metrieken en bepaal normwaarden’ komen verder aan de orde omdat hierin de doel-middel relaties het meest expliciet tot uiting komen. In hoofdstuk 6 is aangegeven dat deze relaties van belang zijn voor de aansturing van het proces.

Tokheim RPS

Tijdens de RPS-opdracht begon men na het formuleren van het meetdoel met het opstellen van vragen. Deze activiteit leverde bij eerste uitvoering echter weinig op. De ontwikkelaars hadden alle ervaring met GoalQuestionMetric-aanpak. Toch hadden ze moeite om vragen over Maintainability van Omega op te stellen. Dit was te abstract. Daarom is de activiteit ‘definieer vragen en hypotheses’ herhaald, maar dan door concreter naar het te beoordelen product te kijken. Hierbij is gebruik gemaakt van bestaand commentaar op de modules. Dit commentaar werd geleverd door ontwikkelaars tijdens een sessie waarbij de kwaliteit van de modules werd vergeleken. Er werden ongeveer 50 opmerkingen gemaakt over de vijf Omega-modules. Deze opmerkingen zijn door de onderzoeker geordend in overleg met twee ontwikkelaars. Dit resulteerde in 9 categorieën, waarvan hieronder drie voorbeelden worden gegeven:

- De omvang van een functie mag niet te groot zijn maar ook niet te klein.
- De code dient een goede lay-out te hebben (formatting).
- Variabelen moeten adequate namen hebben.

Op basis van deze categorieën zijn vragen en hypotheses geformuleerd. Dit betekent voor de eerste van bovenstaande vragen dat er een vraag werd gesteld: ‘wat is de juiste omvang van een functie?’ De bijbehorende hypothese werd bepaald door de ontwikkelaars naar hun mening te vragen. In dit geval kwam daaruit: functiegrootte moet tussen 0 en 60 lines of code

zijn. Dit is gebaseerd op het idee dat onderhoudbare code onder andere betekent dat de code goed is te analyseren. Ten aanzien van analyseerbaarheid speelt dan een rol dat de functie in één oogopslag, te overzien is, dus bijvoorbeeld in één keer op een beeldscherm wordt gepresenteerd. De omvang van een functie die op een beeldscherm past is ongeveer 60 lines of code.

De les die uit deze beoordeling kan worden getrokken is dat er bij het aansturen van de activiteiten ‘vragen en hypotheses’ en ‘selecteer metrieken en bepaal normwaarden’ met referenties gewerkt moet worden. In het geval van Tokheim RPS is deze referentie op basis van het commentaar van de ontwikkelaars opgesteld. Deze les is meegenomen tijdens het inrichten van de beoordelingen bij Océ en Cap Gemini.

Cap Gemini
Tijdens de opdracht bij Cap Gemini zijn vragen opgesteld op basis van de gestelde meetdoelen. Hierbij is gekeken naar relevante onderwerpen in de literatuur. Aldus zijn vragen opgesteld, waaronder de volgende:
- Is het systeem goed te analyseren?
- Is het systeem testbaar na het uitvoeren van wijzigingen?
- Wordt het onderhoudsproces voldoende ondersteund?

Deze en andere vragen zijn gereviewd door een auditor van Cap Gemini. De volgende stap betrof het opstellen van de checklistitems. Dit is in feite het selecteren van metrieken voor de gestelde vragen. Dit is gedaan door eerst na te gaan welke items uit de reeds bestaande checklist de vragen afdekken. Vervolgens is voor de openblijvende vragen gezocht naar nieuwe items. Hierbij zijn checklists uit de literatuur gebruikt. Uiteindelijk zijn voor de audit 7 nieuwe checklists opgesteld die tezamen 140 items betreffen.

Océ
Bij aanvang van de opdracht bij Océ waren er al meetdoelen, (meet)vragen en metrieken gedefinieerd. Deze waren niet ontwikkeld tijdens het project binnen Océ, maar overgenomen van een ander bedrijf dat Maintainability van code wilde meten en daarvoor de GQM-aanpak had gevolgd. Omdat onduidelijk was, wat er precies onder de vragen werd verstaan, is de activiteit ‘bepaal meetvragen’ opnieuw uitgevoerd. Dit is gedaan door een interview te organiseren waarbij een groep engineers aangaf welke vragen met de beoordeling van Maintainability en Reliability (de meetdoelen) beantwoord moesten worden. Het interview werd gehouden aan de hand van een lijst met factoren die in de literatuur wordt gegeven als zijnde factoren die van invloed zijn op Maintainability en Reliability. Voor Maintainability bevatte deze lijst 44 factoren, voor Reliability 24 factoren.

Door deze lijst voor te leggen aan de ontwikkelaars ontstond er discussie die leidde tot een aantal vragen die de ontwikkelaars beantwoord wilden zien. Voor Maintainability zijn

uiteindelijk 20 vragen en voor Reliability 12 vragen geformuleerd. Tijdens het interview is de ontwikkelaars gevraagd om per vraag het belang aan te geven. Hierdoor is per vraag een gewicht, op een schaal van 1 tot 3, toegekend. Na afloop van het interview zijn de resultaten voorgelegd aan de geïnterviewde ontwikkelaars. Op basis van het commentaar is een definitieve verzameling vragen opgesteld. Voor Maintainability bestond deze lijst uit 12 vragen, voor Reliability uit 6 vragen.

De projectgroep 'statisch testen' had vooraf al gedefinieerd welke metrieken gehanteerd moesten worden. Er vond dus feitelijk geen metriekselectie plaats. Er werd echter een consistency check uitgevoerd om te bepalen welke van de codemetrieken de opgestelde vragen afdekken. Zo werd duidelijk dat een aantal vragen niet werd afgedekt door metrieken. Met name voor vragen rondom Reliability werden weinig codemetrieken gevonden. Deze constatering leidde tot bijsturing van het proces. Er waren twee mogelijkheden: of de beoordeling werd ingeperkt door de vragen rondom Reliability weg te laten, of er werden andere metrieken geselecteerd om zo te proberen Reliability alsnog af te dekken. Tijdens de opdracht werd gekozen voor inperking: er werd verder alleen gefocust op Maintainability.

Uit de gang van zaken bij deze drie opdrachten wordt duidelijk dat het operationaliseren van een doel goed mogelijk is door vragen en metrieken op te stellen. Hiervoor moeten wel referenties worden gebruikt. Het proces blanco ingaan had onvoldoende effect. De referenties zelf kunnen zijn opgesteld voor het specifieke proces zoals bij RPS, of worden overgenomen uit de literatuur, zoals in het geval van Océ en Cap Gemini.

Strategie

Er is sprake van twee verschillende strategieën tijdens de drie opdrachten. In het geval van de beoordeling bij Cap Gemini was er sprake van een evolutionaire strategie. Bij Océ en Tokheim RPS wordt een lineaire iteratieve strategie gevolgd.

De strategie bij Cap Gemini duiden we aan als evolutionair omdat de verbetering middels het opstellen van een nieuwe checklist door de organisatie gezien wordt als een stap naar een uiteindelijke beoordeling. De auditors bij Cap Gemini gaan er vanuit dat er per definitie geen algemeen geldende normen over softwareproducten zijn te geven. Om toch meer houvast te krijgen, ondernam men de verbeteringsstap. Startpunt voor het herontwikkelen van de Maintainability audit was de bestaande audit. Doel van de verbeteringsstap was de audit beter te structureren en de vraagstelling te verbeteren door vragen eenduidiger te formuleren.

Dat er bij Océ en Tokheim sprake is van een lineair iteratieve aanpak komt omdat er top-down wordt gewerkt (eerst werden de doelen bepaald, vervolgens de vragen en daarna de te toe te passen metrieken). Men wilde het proces in principe in één keer doorlopen.

Participatie
In hoofdstuk 6 zijn vier factoren genoemd die de participatie van mensen in het beoordelingsproces beïnvloeden. Het betreft: management commitment, motivatie, het gebruik van tools en het aanbieden van referenties. Op deze plaats bekijken we in hoeverre deze vier factoren een rol speelden tijdens de drie beoordelingen.

Management commitment – in één van de drie beoordelingen is expliciet sprake van management commitment ten aanzien van de beoordeling, namelijk tijdens de opdracht bij Océ. Het management van deze organisatie had onderkend dat de kwaliteit van software belangrijk is en dat er daarom een goede beoordeling moest worden ontwikkeld. Vanuit dit commitment is de projectgroep statisch testen voortgekomen. Daardoor was er ook ruimte voor het doen van experimenten om de geschiktheid van de voorgestelde metrieken te bepalen.

Bij de twee andere beoordelingen heeft het hogere management zich niet uitgesproken over het belang van softwarebeoordelingen. In deze gevallen zijn het individuen in de organisatie die zich sterk maken voor een beoordeling omdat zij het belang ervan inzien. Bij Tokheim RPS zagen we dat in de bereidheid van de afzonderlijke ontwikkelaars om mee te werken. Bij Cap Gemini ging het om een individuele beoordelaar die aandrong om in de verbetering van de audit te investeren. In beide gevallen bleek het toch mogelijk om de beoordeling verder te ontwikkelen zonder expliciet management commitment. Hiermee wordt duidelijk dat management commitment weliswaar bevorderlijk is, maar dat het geen absoluut noodzakelijke voorwaarde is.

Motivatie – in alle drie de beoordelingen is de motivatie van betrokkenen van belang. Het opstellen van meetdoelen, vragen en metrieken verloopt niet vanzelf en vereist inzet van de mensen. Tijdens de Océ-opdracht werd de invloed van de motivatie het meest zichtbaar omdat daar twee groepen ontwikkelaars bij de inrichting van de beoordeling betrokken waren. Ondanks de scepsis van beide groepen over de bruikbaarheid van softwaremetrieken om kwaliteit van software te bepalen, was er één groep bereid om te investeren in de ontwikkeling van de beoordeling. Deze groep was duidelijk meer gemotiveerd dan de andere groep. Deze groep manifesteerde zich dan ook veel actiever tijdens het formuleren van vragen, het leveren van feedback op tussentijdse resultaten en het meedenken tijdens de beoordeling. De als ‘minder gemotiveerd’ te karakteriseren groep deed weliswaar mee aan de interviews, maar leverde feitelijk nauwelijks een bijdrage. Zo kwam ondanks herhaalde oproepen tot terugkoppeling geen reactie. Het was vervolgens bijna onmogelijk om de kwaliteit van de software te bepalen, die door deze groep geleverd werd.

Toepassen van tools – in elk van de beoordelingen zijn geautomatiseerde tools toegepast tijdens de uitvoering. Bij Océ en Tokheim RPS betrof het static analysis tools waarmee aan code wordt gemeten. Bij Cap Gemini ging het om een geautomatiseerde checklist waarmee

antwoorden op vragen worden gegenereerd. Voor de eerste twee beoordelingen stond bij voorbaat de implementatie door middel van tools vast, bij de derde niet. Het is daarom interessant om de invloed van het tool op het te organiseren proces te bekijken.

Er hebben verschillende overwegingen meegespeeld om de audit checklists bij Cap Gemini te automatiseren. De eerste is dat de checklists veel items omvatten. Door deze items in een spreadsheet te implementeren hoefden mensen alleen te antwoorden waarna de score automatisch werd gegenereerd. Een ander voordeel van de geautomatiseerde checklist was dat er op twee manieren een score werd bepaald, namelijk naar meetvraag en naar onderdeel van het softwareproduct. Het ordenen naar vraag ligt voor de hand. Het ordenen naar onderdeel komt voort uit de door auditors gewenste manier van beoordelen, namelijk om per productonderdeel een oordeel te geven. De implementatie van items in een spreadsheet maakte het mogelijk om beide varianten zonder extra werk door te rekenen. Deze 'feature' van het tool speelde een belangrijke rol bij het accepteren van de beoordeling door de beoordelaars.

Een andere reden om een spreadsheet te gebruiken was het idee dat het aldus mogelijk was om de rekenregels die ten grondslag liggen aan het criteriamodel en de normwaarden expliciet te maken. We verwachtten dat door het gebruik van een spreadsheet de personen naar eigen inzicht deze regels en normwaarden zouden aanpassen. Het op deze manier toepassen van een checklist doet recht aan het gegeven dat men een beoordeling moet toesnijden op de specifieke situatie.

Hetzelfde idee werd ook toegepast tijdens de opzet van de beoordeling bij Océ. Ook hier was het uitgangspunt dat mensen in de organisaties het beste in staat zijn om criteria en normwaarden aan het product (bij) te stellen. De geautomatiseerde checklist bij Océ werd opgesteld om het proces rondom de activiteiten 'opstellen van vragen en hypotheses' en 'selecteer metrieken en normwaarden' te ondersteunen.

Tijdens de introductie van de geautomatiseerde checklists werd in beide organisaties de mogelijkheid om criteria aan te passen expliciet genoemd en bepleit. In beide organistaties is dat echter niet gebeurd. Men nam het bediscussieerde model over, stelde wel vragen, maar paste de criteria en normwaarden niet aan. Het beoogde effect van de geautomatiseerde ondersteuning, namelijk dat mensen zelf normen en criteria gaan bijstellen, werd dus niet bereikt. Wellicht dat het herhaald propageren van deze aanpak wel tot aanpassingen had geleid.

Aanbieden van referenties – bij de behandeling van de processtructuur in paragraaf 7.2.2 is deze factor al aan de orde gekomen. Om de overgang van doel naar middelen te maken, bleek het noodzakelijk om referenties te gebruiken. Op deze manier werden aan de mensen alternatieven geboden, waaruit ze konden kiezen. De mensen werden hierdoor intensiever bij

het proces betrokken. Het aanbieden van referenties is daarmee een belangrijke factor voor partipatie.

7.2.3 Afwegen van doel en middelen

Ten aanzien van het derde aspect van het ontwerp, namelijk het afwegen van doel en middelen, wordt eerst vastgesteld waar de afweging in het proces plaatsvond. Dit heeft betrekking op de richtlijn 'onderken expliciet de afwegingsmomenten in het proces'. Vervolgens gaan we in op een richtlijn die actief is toegepast, namelijk 'het expliciteren van doel en middelen'. Bij deze richtlijn hebben we ons gericht op het expliciteren van metrieken en daarbinnen op het ontsluiten van metrieken middels de metriekendatabase, zoals geïntroduceerd in hoofdstuk 6. Tijdens de opdrachten bij Tokheim RPS en Cap Gemini zijn hiermee ervaringen opgedaan.

Onderken expliciet de afwegingsmomenten in het proces

Voor elk van de drie opdrachten hebben we geprobeerd om vast te stellen waar in het proces de afweging van doel en middelen plaatsvond.

Tijdens de processen bij Tokheim RPS en Océ vond de afweging plaats op het niveau 'organisatiedoel – meetdoel'. Met het formuleren van het meetdoel werd duidelijk dat er met codemetrieken zou worden gemeten. Ook het element 'object' van het meetdoel wijst hierop: er wordt gesproken van (source)code. Verder spraken de deelnemers aan het proces steeds over het toepassen van codemetrieken.

De volgorde waarin de afwegingsmomenten aan de orde komen hangt samen met de gevolgde strategie. Eerder is aangegeven dat in de opdrachten bij Océ en Tokheim een lineair iteratieve strategie gevolgd wordt. De afwegingsmomenten komen na elkaar aan de orde: eerst meetdoel-vraag en vervolgens vraag-metriek niveau. Er wordt hierbij gewerkt naar het vinden van de juiste codemetrieken. De gang van zaken rondom de Maintainability audit is wat dit betreft meer open. In het begin van het proces lagen alle mogelijkheden nog open. Zo werd gesproken over het toepassen van codemetrieken, inspecties en het gebruik van checklists. De keuze voor het toe te passen middel werd uiteindelijk gemaakt tijdens het bepalen van meetvragen én het selecteren van de metrieken voor deze vragen. De afweging vond daarmee plaats op twee doel-middel niveaus, namelijk: meetdoel-vraag en vraag-metriek en terug gaan naar een voorgaand niveau stond open.

Expliciteren van doelen en middelen: metriekendatabase

In hoofdstuk 6 zijn verschillende attributen gegeven waarmee metrieken in een ervaringsdatabase opgeslagen kunnen worden. In deze paragraaf bekijken we welke van deze attributen een rol speelden tijdens het afwegingsproces in de opdrachten.

De metriekendatabase is tijdens twee opdrachten toegepast. In de opdracht bij Tokheim RPS is de database gebruikt om alternatieven voor de bestaande metrieken te genereren. Tijdens de opdracht bij Cap Gemini is de database gebruikt om checklists op te stellen. In beide gevallen is eerst bepaald welke kwaliteitskarakteristieken en onderwerpen van het product (object) van belang zijn. Hiervoor is gekeken naar de opgestelde meetdoelen (zie paragraaf 7.2.1).

Het meetdoel bij Tokheim RPS indiceert dat modules het object van beoordeling zijn en dat de quality focus op Maintainability ligt. Tijdens de opdracht bij Tokheim RPS werden er op deze basis metrieken uit de database geselecteerd. Dit werd gedaan door een query uit te voeren waarmee alle metrieken werden geselecteerd die het attribuut (source) code bevatten. Dit leverde 29 metriekomschrijvingen op. De totale database omvatte op dat moment 200 metriekbeschrijvingen. De metrieken werden aan de organisatie voorgesteld als alternatieven voor de al in gebruik zijnde metrieken. Het oordeel van de organisatie was: ‘interessant, maar meten deze metrieken nu beter maintainability?’ Een terechte vraag die echter niet kon worden beantwoord met de bestaande kennis. Er was te weinig informatie over de metrieken. Zo kon er niet worden gegarandeerd dat met de geselecteerde metrieken de Maintainability van Omega-modules beter kon worden bepaald. Een extra probleem was dat er niet aan de Omega-code kon worden gemeten met de geselecteerde metrieken omdat implementatie in het statische analysetool te veel tijd zou kosten.

In het meetdoel van Cap Gemini stond Maintainability eveneens centraal. Het object werd daar echter breder omschreven dan bij Tokheim RPS. Allereerst werd er gesproken over het beoordelen van een informatiesysteem. Men vatte hieronder programmatuur en documentatie. Daarnaast diende er ook naar het onderhoudsproces te worden gekeken. Om de keuze van de onderwerpen te structureren werd tijdens de opdracht onderstaande tabel gebruikt. Deze tabel werd voorgelegd aan de Cap Gemini auditors met de vraag om een keuze te maken welke onderwerpen tijdens een Maintainability-toets aan de orde dienden te komen. De tabel werd ontwikkeld om een overzicht te hebben van de onderwerpen waar men tijdens een Maintainability-beoordeling aan kan meten. Hiervoor zijn bestaande indelingen (Oman e.a., 1991), (Fenton en Pfleeger, 1996), (Hausen en Welzel, 1993) gebruikt.

Tabel 7.3 Overzicht van onderwerpen voor een (Maintainability) beoordeling

Product	Omgeving	Proces	Hulpmiddelen
Programmatuur (source code) van het product) Documentatie (van het product) Ontwerp (van het product) Product als werkend systeem	Interfaces van het product met andere systemen – bijvoorbeeld met bestaande infrastructuur Taken van mensen die product gebruiken – 'system fit' Gegevens verwerkt door het systeem	Ontwikkelproces – methoden gebruikt tijdens specificeren en coderen van het product Onderhoudsproces van product – uitvoeren van onderhoud, beheer van onderhoud (change management)	Mensen betrokken bij het product – hun kennis en expertise

De auditors bepaalden vervolgens dat de volgende onderwerpen tijdens een Maintainability audit aan de orde moesten komen: programmatuur, documentatie, interfaces met andere systemen, onderhoudsproces en mensen betrokken bij het onderhoudsproces. Vervolgens is de metrieken database geraadpleegd en zijn metrieken geselecteerd op basis van het attribuut 'object'. Dit resulteerde in een aantal codemetrieken en vooral checklist items die vervolgens werden geordend conform de opzet van de audit, namelijk naar (meet)vraag en onderwerp (zie paragraaf 7.2.2).

De geselecteerde metrieken dekten echter nog onvoldoende een aantal van de (meet)vragen en onderwerpen. Soms werd voor een meetvraag maar 1 item geselecteerd. In vergelijking met andere vragen, waarvoor bijvoorbeeld 10 items werden geselecteerd, werd dit als te mager beschouwd. Verder nuanceerden de beoordelaar/auditors diverse meetvragen. Dit betrof dan opmerkingen in de trant van: 'om alleen te kijken naar volledigheid en structuur van documentatie is te beperkt. Je moet ook kijken naar accuraatheid en inzichtelijkheid'. Dit leidde tot een tweede ronde van metriekselectie, waarbij niet de metriekendatabase maar checklists uit de literatuur werden geraadpleegd. Op deze manier werden de checklists alsnog aangevuld.

Het voorgaande laat zien dat het selecteren van metrieken uit de metriekendatabase niet vlekkeloos verliep. Dit hangt samen met de beperkte inhoud van de database alsmede het onvoldoende kunnen onderbouwen van de waarde van een concreet metriekenvoorstel.

Beperkte inhoud van de database – tijdens de opdrachten bleek dat de selecties te mager waren. Er moest naar andere referenties worden gekeken om tot een redelijk voorstel te komen. Dit laat zien dat men voor een succesvolle selectie afhankelijk is van de inhoud van de database. Door meer metriekvoorstellen in de database op te nemen, is hier in principe sprake van een tijdelijk probleem.

Onderbouwing van het voorstel – tijdens de opdrachten bleek dat een metriekenvoorstel moeilijk te onderbouwen was. Dat een metriek meet aan code of aan documentatie is onvoldoende om mensen er van te overtuigen de metriek te gaan gebruiken. Dit probleem speelt ook ten aanzien van de andere attributen die in de metriekendatabase werden bijgehouden, zoals: perspectief', 'moment in de levenscyclus van het product waarop de beoordeling plaatsvindt' en 'soort meting'. Om dit probleem aan te pakken wordt voorgesteld om metrieken in een ervaringsdatabase ook te ordenen op basis van de meetvragen. Vragen hebben een logische relatie met metrieken en worden in de processtructuur voorafgaand aan metrieken geformuleerd. Dit leidt tot een aanpassing van het datamodel zoals gepresenteerd in hoofdstuk 6. De entiteit 'vraag' wordt namelijk aan het model toegevoegd.

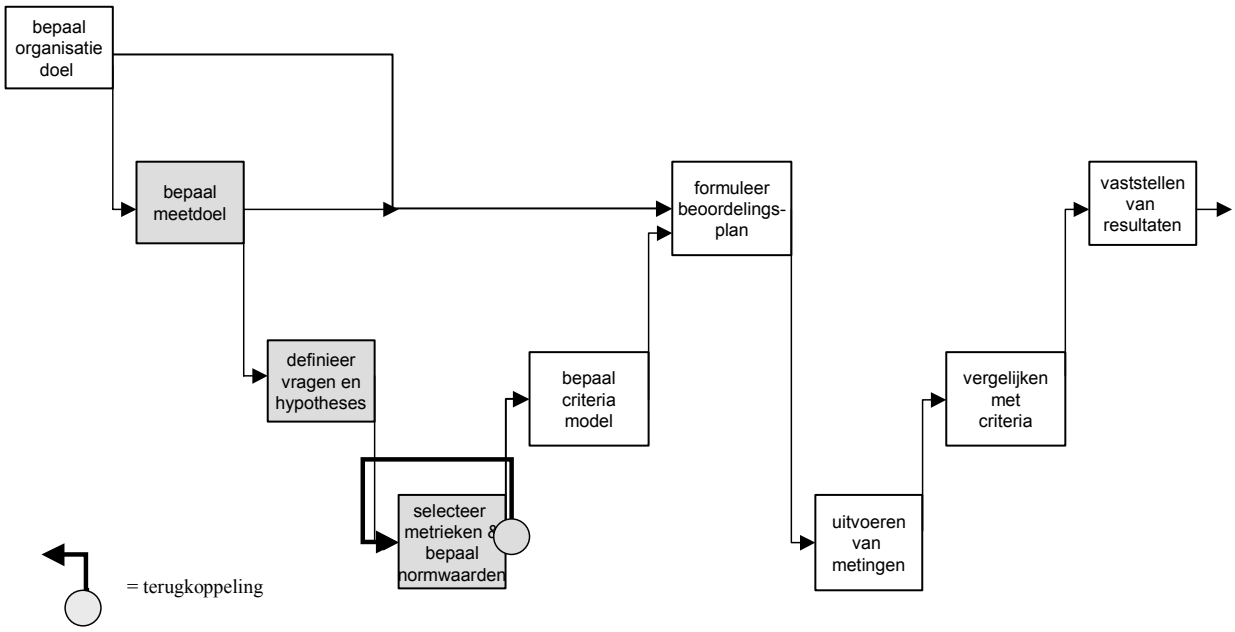
Het toevoegen van nieuwe metriekbeschrijvingen en het wijzigen van het datamodel van de database maakt dat er verder onderzoek ten aanzien van de metriekendatabase nodig is. Het afstemmen van de metrieken op een specifieke meetsituatie is ingewikkelder dan het in eerste instantie leek; zie ook de aanbevelingen in hoofdstuk 8.

7.2.4 Terugkoppeling en bijsturing

Bij de behandeling van het vierde ontwerpaspect wordt eerst ingegaan op de richtlijn 'onderken expliciet terugkoppelmomenten in het proces'. Hiervoor stellen we vast waar in het proces van de uitgevoerde opdrachten sprake was van terugkoppeling. Vervolgens gaan we in op de richtlijn die aangeeft dat problemen in het proces geïdentificeerd en de oorzaken ervan bepaald moeten worden om bijsturingsinformatie van een voldoende 'volwassenheidsniveau' te genereren.

Terugkoppelmomenten

Tijdens de opdrachten bij Océ en Tokheim RPS was sprake van terugkoppeling tijdens het proces. In beide gevallen gebeurde dit naar aanleiding van de uitgevoerde metriekselectie. De metriekselectie werd ter discussie gesteld: of deze de te meten karakteristiek (Maintainability) goed kon voorspellen. De terugkoppeling betrof daarmee in beide gevallen het vraag-metriek niveau en had betrekking op de activiteit 'selecteer metrieken en normwaarden'. Er werd niet overwogen om een daaraan voorafgaande activiteit opnieuw uit tevoren door bijvoorbeeld andere vragen te gaan stellen. Onderstaande figuur geeft het terugkoppelmoment.



Figuur 7.2 Terugkoppelmoment in het proces bij Océ en Tokheim RPS

In beide gevallen is de terugkoppeling uitgevoerd door de metriekresultaten te vergelijken met de opinies van ontwikkelaars. Deze meningen werden gescoord aan de hand van een gestructureerde vragenlijst met gesloten antwoordmogelijkheden. De vragenlijsten waren afgestemd op de betreffende softwareproducten en pretendeerden de Maintainability van de Boston/Dac en Omega software te meten. Voor de analyse bij Tokheim RPS wordt verwezen naar Punter (1999). Hieronder wordt ingegaan op de analyse bij Océ.

Tijdens de opdracht bij Océ werden de metriekwaarden vergeleken met historische data omtrent de inspanning voor de ontwikkeling van de modules. Dit is gebeurd vanuit de gedachte dat een module die veel inspanning heeft gekost, ook in de toekomst veel inspanning zal blijven kosten (van der Zwan en Punter, 1999). De vergelijking is gebaseerd op drie variabelen, namelijk: metriekwaarden, inspanningsgegevens en opinies van de ontwikkelaars. Voor elke variabele werd een ranking van de modules gemaakt op een ordinale schaal. De drie volgordes zijn weergegeven in de onderstaande tabel.

Tabel 7.4 Kwaliteit van Boston modules (bij Océ) gebaseerd op drie variabelen

	Uitkomst gebaseerd op:		
Kwaliteit	Metrieken	Inspanning per module	Engineer opinie
Hoog	module 1	module 8	module 2
▲	module 2	module 5	module 1
|	module 3	module 4	module 5
|	module 4	module 1	module 6
|	module 5	module 9	module 7
|	module 6	module 7	module 8
|	module 7	module 2	module 3
▼	module 8	module 6	module 4
Laag	module 9	module 3	module 9

Deze tabel laat drie verschillende volgordes zien: elk van de variabelen geeft een eigen volgorde van de mate van onderhoudbaarheid van de modules. Op basis van deze resultaten is door de projectgroep binnen Océ geconstateerd dat de geselecteerde set metrieken niet voldoet om Maintainability vast te stellen. Men vond de relatie tussen de codemetrieken en de twee andere variabelelen té weinig samenhangend om codemetrieken als voorspeller van Maintainability te kiezen.

Meten aan het proces en problemen en oorzaken bepalen

Meten – in beide opdrachten is de (on)tevredenheid van de ontwikkelaars over toegepaste metrieken een belangrijke indicator voor de constatering of het proces 'goed verloopt'. In de Tokheim RPS opdracht zagen we dit duidelijk aan de meningen van ontwikkelaars. Dit is in hoofdstuk 4 beschreven. De geconstateerde ontevredenheid vormde de aanleiding voor de opdracht die in dit hoofdstuk is beschreven. Ook hier is tevredenheid een belangrijke indicator. Immers, de nieuwe metriekselectie wordt getoetst aan de hand van de meningen van de ontwikkelaars.

Problemen – de processtructuur, zoals opgesteld in figuur 7.1 laat zien dat er in de cases bij Tokheim RPS en Océ drie activiteiten zijn uitgevoerd, namelijk: 'bepalen meetdoel', 'definiëren vragen en hypotheses' en 'selecteren metrieken en normwaarden'. In principe kunnen problemen zich voordoen in elk van deze activiteiten. Echter, tijdens de opdrachtuitvoering richtte zich dit maar op één activiteit: de metriekselectie. De vraag is dan ook waarom de problemen bij de metriekselectie worden gelegd? Dit komt vooral door de scope die de beoordelingen hadden bij de aanvang. In beide gevallen waren het object en de kwaliteitskarakteristieken al bepaald, namelijk: beoordelen van (source)code en Maintainability. Hierdoor hadden de mensen in beide organisaties duidelijk voor ogen dat er bepaalde metrieken moesten worden geselecteerd. Ondanks het feit dat tijdens de opdrachten de activiteiten 'bepaal meetdoel' en 'definiëren van vragen en hypotheses' alsnog werden uitgevoerd, bleef de aandacht gericht op de metriekselectie. We leren hieruit dat de start van een traject van groot belang is voor de inrichting van het proces én de terugkoppeling. Een

beoordelaar in de rol van moderator moet de kracht en de durf hebben om kritisch naar het waarom van keuzes in voorgaande activiteiten te vragen.

Oorzaken – om de oorzaken van problemen in een proces te achterhalen is het van belang om oorzaak-gevolg relaties te kennen. In hoofdstuk 6 is aangegeven dat kennis hierover nog grotendeels ontbreekt. Er is daarom voorgesteld om root-cause modellen per activiteit van de processtructuur op te stellen. Om hier een bijdrage aan te leveren is voor de opdrachten bij Océ en Tokheim RPS vastgesteld wat de oorzaken van de problemen waren. We hebben ons hierbij gericht op twee activiteiten: 'definiëren van vragen en hypotheses' en 'selecteren metrieken en normwaarden'.

Oorzaken van problemen die zich tijdens het 'definiëren van vragen en hypotheses' voordeden, waren:

- De beperking van mensen om zich in te leven in het product en vereiste kwaliteit – dit zagen we met name tijdens de opdracht bij Tokheim RPS toen bleek dat de ontwikkelaars (ondanks GQM-ervaring!) geen vragen over het product konden formuleren. De vraag wat men in het algemeen onder Maintainability verstaat, is te abstract. Door het daarna aan de hand van het concrete product te vragen, konden alsnog vragen worden geformuleerd.
- Te beperkt gezichtspunt – tijdens beide opdrachten werden ontwikkelaars uigenodigd om vragen te formuleren. De uitvoering van deze activiteit verbeterde toen ook de quality engineers werden geraadpleegd. Betrokkenen merkten op dat er betere vragen werden gesteld. Bij Tokheim RPS is daarom overwogen om ook de afnemers van de software te consulteren. Het betrekken van meerdere actoren leidt wellicht tot conflicterende vragen (zie doelformulering in hoofdstuk 6), maar het stimuleert het opstellen van vragen wel.

Oorzaken van problemen die zich tijdens het 'selecteren metrieken en normwaarden' voordeden waren:

- Ontbreken van een meetmodel – de normwaarden van een aantal metrieken werd geschat en niet beargumenteerd. Er ontbrak een meetmodel waarin beredeneerd wordt waarom bijvoorbeeld nesting van 4 nog wel en nesting van 5 niet meer voldoet.
- De gehanteerde meetconcepten – bij het vaststellen van normwaarden werd er in beide opdrachten vanuit gegaan dat er minimum- en maximumwaarden bestaan. Een acceptabele waarde ligt dan tussen het minimum en maximum. Dit veronderstelt veelal een lineair verband tussen metriekwaarden. Dit is echter niet het geval.
- Meetprogramma's – in paragraaf 7.2.2 is het gebruik van referenties gepropageerd om de overgang van doel naar middelen mogelijk te maken. Er schuilt echter ook een gevaar in, namelijk dat niet verder wordt nagedacht over de vraag 'of de metrieken voor Maintainability in de context van organisatie X, ook de juiste zijn voor een soortgelijke beoordeling bij bedrijf Y'. Softwareproductkwaliteit is immers contextafhankelijk. In de opdrachten komen we dit aspect tegen bij Océ waar in eerste instantie een compleet

model met meetdoelen, vragen en metrieken was overgenomen van een andere organisatie, zonder dat dit was toegespitst op de context van Océ. Tijdens de opdracht is dit alsnog gedaan.
- Complementariteit codemetrieken en checklist benadering – tijdens de opdrachten bij Tokhheim RPS en Océ zijn codemetrieken geselecteerd. Er is voorgesteld om ook checklist items toe te passen. Uiteindelijk zijn deze niet toegepast. In beide gevallen had dit tot gevolg dat de meetvragen beperkt werden afgedekt en dat dit met het complementair toepassen van checklist items beter was gebeurd.

Op basis van het voorgaande concluderen we dat een beoordelingsproces expliciet en gestructureerd moet worden doorlopen om de besturingsinformatie op niveau 3 te krijgen. De keuzes in het proces moeten expliciet zijn geformuleerd en de activiteiten moeten daadwerkelijk zijn uitgevoerd. Pas dan zijn problemen en oorzaken in het beoordelingsproces goed te detecteren.

7.3 Evaluatie van de tevredenheid over het ontwerp

Om de tevredenheid over het ontwerp, dat is opgesteld in hoofdstuk 6 vast te stellen, zijn we uitgegaan van de vier beschreven probleemgebieden, namelijk: onvoldoende doelformulering, onvoldoende aansturing, onvoldoende afweging van doel en middelen en onvoldoende terugkoppeling en bijsturing. Het bepalen van de tevredenheid ten aanzien van ons ontwerp op deze gebieden is echter niet eenvoudig. Er zijn voor elk gebied meerdere, soms geheel verschillende, vragen te stellen. Zo kan ten aanzien van de onvoldoende aansturing worden gevraagd naar: ‘is er sprake van aansturing?’, ‘levert het project het juiste (of gewenste) resultaat op?’, ‘verloopt het proces goed?’. Daarnaast bestaat het gevaar dat tevredenheid of ontevredenheid slaat op de resultaten van een beoordelingsproces in plaats van op het ontwerp. Tenslotte speelt bij het meten van de tevredenheid of de vragen aansluiten op de belevingswereld van de geïnterviewden (Bartelds e.a., 1989). Al deze overwegingen hebben meegespeeld bij het stellen van de volgende vragen ten aanzien van het door ons voorgestelde ontwerp:
- Is het doel van de beoordeling duidelijk?
- Is het gevolgde proces duidelijk?
- Worden er geschikte middelen geselecteerd?
- Wordt er op tijd ingegrepen?

De tevredenheid over ons ontwerp is aan de hand van voorgaande vragen tijdens de opdracht bij Cap Gemini vastgesteld. Dit is eveneens deels bij Océ gedaan.

Is het doel van de beoordeling duidelijk?
Op de vraag aan betrokkenen of het doel duidelijk is geformuleerd met de meetdoel en goal template formulering is het antwoord steeds positief. Zo vond men bij Cap Gemini de

combinatie tussen kwaliteitskarakteristiek en de vermelding van het object verhelderend. Mensen bij Océ vonden het verhelderend om te zien dat er met hun beoordeling niet één doel maar twee meetdoelen moesten worden uitgewerkt: Maintainability én Reliability. De mensen vonden het minder duidelijk hoe men dit meetdoel moest uitwerken. Bij Cap Gemini speelde de vraag hoe men een meetdoel kon omzetten in een goede checklist.

Is het gevolgde proces duidelijk?
In beide organisaties werd het gevolgde proces als gestructureerd ervaren. Het was voor de mensen duidelijk welke stappen werden genomen en wat de geplande resultaten waren. De geïnterviewden waarderen de aanpak waarbij vanuit meetdoelen, via vragen, metrieken worden opgesteld. Ook de ondersteuning van referenties hierbij werd positief ervaren. In de opdracht bij Cap Gemini wordt dit verwoord door de opmerking van de auditor dat hij de opgestelde vragen ervaart als een goede afspiegeling van hetgeen hij te weten wil komen tijdens een audit. De opgestelde metrieken worden als bruikbaar ervaren: ze geven antwoord op de gestelde vragen. De auditor van Cap Gemini is gecharmeerd van de relatie tussen de checklist items en de beantwoording van de vragen.

Worden er geschikte middelen geselecteerd?
In beide opdrachten heeft de beantwoording zich gericht op de tevredenheid over de geselecteerde metrieken. In beide organisaties is men redelijk tevreden over het opgeleverde advies. Hierbij past de opmerking dat het ambitieniveau ten aanzien van de beoordeling in beide gevallen oorspronkelijk hoger lag dan gerealiseerd kon worden. Desondanks hebben de geïnterviewden op basis van het doorlopen proces het idee dat de best mogelijke verzameling van middelen is toegepast. Bij Cap Gemini wordt dit verwoord als: ‘we hebben met dit beoordelingsproces gezien dat we de state-of-the-art zijn met betrekking tot beoordelen ... we kunnen kennelijk moeilijk ‘hardere’ normen in onze audits hanteren’. De oorspronkelijke ambities zijn dus getemperd, maar men accepteert de situatie.

Wordt er op tijd ingegrepen?
De projectgroep bij Océ heeft besloten om vooralsnog niet op grotere schaal met codemetrieken te werken: het levert onvoldoende op. Op de vraag of er op tijd werd ingegrepen in het proces, gaven de mensen aan dat het experiment nuttig was geweest en inzicht heeft opgeleverd dat codemetrieken nog niet voldoende voor de organisatie opleveren.

7.4 Conclusies

In dit hoofdstuk is het ontwerp voor doelgericht beoordelen (deels) geëvalueerd. Dit is gebeurd door te kijken naar de toepasbaarheid van het ontwerp en naar de tevredenheid van mensen die met het ontwerp hebben gewerkt.

Ten aanzien van de processtructuur zijn vooral de activiteiten ‘formuleren meetdoelen’, ‘definiëren vragen en hypotheses’ en ‘selecteren metrieken en normwaarden’ aan de orde gekomen. Tezamen met de richtlijn ‘operationaliseren van doelen en het gebruik van measurement goal template’ bleek dit deel van het ontwerp goed toepasbaar. Het volgen van de richtlijn ‘expliciteren van middelen tijdens afwegingsproces’ laat zien dat het toepassen van een ervaringsdatabase nog niet zo eenvoudig is. We hebben dan ook een voorstel tot verbetering gedaan namelijk het expliciet toevoegen van informatie over de relatie tussen meetvraag enerzijds en metriek anderzijds.

Over de andere, niet actief geverifieerde richtlijnen maken we de volgende opmerkingen:
- in de bekeken beoordelingen is het mogelijk om partijen te onderkennen in de vier beschreven standaardrollen in een beoordeling,
- van de in hoofdstuk 6 onderkende strategiën troffen we de lineair iteratieve en de evolutionaire strategie aan. We ontdekten in de bestudeerde gevallen een relatie tussen strategie en de afwegingsmomenten in het proces. In een lineaire strategie ligt dit afwegingspunt op een hoger doel-middel niveau dan in een evolutionaire strategie. In deze laatste strategie wordt het afwegingsmoment uitgesteld en is afweging op lagere niveaus mogelijk,
- ten aanzien van terugkoppeling constateren we dat het beoordelingsproces expliciet en gestructrureerd moet zijn uitgevoerd, wil er sprake zijn van besturingsinformatie van voldoende niveau.

De evaluatie van het ontwerp, zoals besproken in dit hoofdstuk, laat zien dat het ontwerp goede perspectieven biedt voor het gestructureerd uitvoeren van beoordelingsprocessen voor software.

8. Samenvatting, conclusies en aanbevelingen

In dit laatste hoofdstuk wordt het onderzoek samengevat en worden er conclusies getrokken. Vervolgens worden er aanbevelingen voor vervolgonderzoek gedaan.

8.1 Samenvatting en conclusies

Er is het afgelopen decennium veel vooruitgang geboekt met het beoordelen van softwareproducten. De ISO 9126-standaard heeft ertoe geleid dat er een begrippenapparaat is voor software-kwaliteit. De ISO 14598-standaard heeft het beoordelingsproces zelf gedefinieerd. Een aantal Esprit-projecten heeft belangrijke concepten opgeleverd, zoals: evaluatiemodule (Scope) en kwaliteitsprofiel (Space-Ufo). Verder zijn er een groot aantal metrieken, checklisten en andere hulpmiddelen ter ondersteuning van het proces gedefinieerd.

Ondanks deze vooruitgang bestaat er in de praktijk ontevredenheid over softwareproductbeoordelingen. Zo komt het voor dat de klant van een beoordeling ontevreden is: hij of zij kan er niets mee omdat de beoordeling antwoord geeft op een verkeerde vraag. Er zijn dan wel beoordelingsactiviteiten uitgevoerd en up-to-date hulpmiddelen toegepast, maar het ontbreekt aan een behoorlijke aansturing van activiteiten, aan een afstemming tussen doel en middelen en aan terugkoppeling om na te gaan of het gestelde doel is bereikt. Kortom: beoordelingen van softwareproducten worden vaak onvoldoende bestuurd. Deze probleemstelling was de aanleiding van ons onderzoek.

In dit onderzoek is de besturingsproblematiek geanalyseerd en is er een ontwerp ter verbetering opgesteld. Hiervoor zijn twee conceptuele modellen opgesteld. Het eerste conceptuele model is het analysekader om softwareproductbeoordelingen te typeren. Het is beschreven in de hoofdstukken 3, 4 en 5. Het tweede conceptuele model is een ontwerp voor het besturen van beoordelingsprocessen (hoofdstukken 6 en 7). Het ontwerp is aangeduid als doelgericht beoordelen. Met deze benadering wordt beoogd om beoordelingsprocessen beter te besturen zodat deze processen resultaten opleveren waarom de opdrachtgever echt vraagt.

8.1.1 Conclusies bij het analysekader

Bij het opstellen van het analysekader is ervan uitgegaan dat het bij een softwareproductbeoordeling om meerdere beoordelingsactiviteiten gaat. Er is gesteld dat beoordelen zich niet beperkt tot het vergelijken met de norm, maar juist ook voorbereidende activiteiten betreft zoals ‘het opstellen van een kwaliteitsmodel’ en ‘het bepalen van de

metrieken'. Er is dan ook geen sprake van één beoordelingsactiviteit, maar van een beoordelingsproces, waartoe verschillende activiteiten behoren.

Proces – in de literatuur zijn verschillende procesbeschrijvingen te vinden. Elke beoordelingsmethode kiest zijn eigen omschrijving van activiteiten en heeft een eigen volgorde van beoordelingsactiviteiten. ISO 14598 biedt een standaard procesbeschrijving. Er zijn echter aanvullingen op mogelijk. Er is dan ook geen algemeen geldende procesbeschrijving te geven. Ook de in dit proefschrift opgestelde processtructuur –die de ISO 14598-beschrijving tracht te verbeteren– zal altijd met één of meer activiteiten kunnen worden aangevuld.

Besturing – tijdens het onderzoek is vanuit een bedrijfskundige invalshoek naar beoordelingsprocessen gekeken. Daarom is juist de besturing van de diverse beoordelingsactiviteiten onderwerp van onderzoek. Bij het uitwerken van dit besturings-perspectief zijn concepten uit de systeemtheorie gebruikt. Op basis hiervan is een analysekader uitgewerkt dat zich richt op vijf aspecten, namelijk:

1. Activiteiten – de beschrijving van activiteiten: wat gebeurt er tijdens het proces?
2. Processtructuur – de samenhang tussen de activiteiten. Dit geeft de volgorde van uitvoering aan.
3. Aansturing van het proces – betreft het inrichten van het proces (welke activiteiten zijn nodig om het doel te bereiken), het toewijzen van middelen aan het proces en het onderkennen van voortgangsbewaking.
4. Afwegen van doel en middelen – dit betreft de afweging die vooraf gaat aan het toewijzen van middelen aan het proces. De kwaliteit en de kwantiteit van de middelen zijn hierbij van belang.
5. Terugkoppelen en bijsturen van het proces – een beoordelingsproces moet worden bijgestuurd als dreigt dat het oorspronkelijk bepaalde doel niet wordt gehaald. Om bij te kunnen sturen is terugkoppeling noodzakelijk: er moet worden geconstateerd dat er 'iets mis gaat' in het proces.

Met het analysekader is bekeken of bestaande beoordelingsmethoden aandacht besteden aan de onderkende aspecten. Dit is gebeurd middels een literatuurstudie. De analyse laat zien dat bestaande beoordelingsmethoden onvoldoende aandacht geven aan de laatste drie aspecten: aansturing, afweging en terugkoppeling. Tijdens de literatuurstudie kwam ook naar voren dat het formuleren van een doel een belangrijk element van de besturing is en dat dit door de bestaande methoden vaak wordt onderschat.

Probleemstelling – deze bevindingen hebben geleid tot de volgende aangescherpte probleemstelling: het beoordelen van softwareproducten leidt veelal tot onbevredigende resultaten, doordat het ontbreekt aan een goede besturing van het beoordelingsproces. Tekortkomingen in deze besturing liggen op de volgende vier probleemgebieden:

1. Onvoldoende aandacht voor het formuleren van een operationele doelstelling, waarin alle betrokken partijen zich herkennen.
2. Onvoldoende inzicht in de activiteiten van het beoordelingsproces en de samenhang daartussen (processtructuur). Daardoor schort het aan een goede aansturing van beoordelingsprocessen.
3. Onvoldoende aandacht voor het afwegen van doel en middelen. Bestaande beoordelingsmethoden onderkennen dit balanceervraagstuk nauwelijks, maar veronderstellen 'vaste' doelen en een oneindige capaciteit aan middelen, zowel in kwantitatief als kwalitatief opzicht.
4. Onvoldoende aandacht voor het terugkoppelen van resultaten, eventueel gevolgd door het bijstellen van doel, middelen en activiteiten.

Deze probleemstelling is getoetst door twee casestudies uit te voeren (zie hoofdstukken 4 en 5). Het betrof hierbij een codemetriek en een vraaggebaseerde beoordeling. Daarmee zijn twee uiteenlopende beoordelingstypen afgedekt. In beide gevallen bleek het analysekader een bruikbaar instrument om problemen met de besturing in beoordelingsprocessen te identificeren. Deze praktijktoets bevestigt dat de genoemde 4 deelproblemen uiterst relevant zijn. Ze zijn daarom het uitgangspunt voor een ontwerp ter verbetering van softwareproductbeoordelingen.

8.1.2 Conclusies bij het ontwerp

Om de problemen rondom het besturen van beoordelingsprocessen aan te passen is een ontwerp opgesteld, bestaande uit een processtructuur en een set richtlijnen. Bij het opstellen van dit ontwerp is ervan uitgegaan dat beoordelen doelgericht moet gebeuren.

De processtructuur is een overzicht van uit te voeren activiteiten en de relaties daartussen. Het is een aanpassing van de structuur zoals opgesteld in de ISO 14598-standaard. In de nieuwe structuur zijn expliciet doel-middel relaties onderkend. Voor de uitwerking hiervan is het GoalQuestionMetric-principe toegepast. Het gehele proces omvat negen activiteiten. De beschrijving is te vinden in hoofdstuk 6.

De processtructuur alleen is onvoldoende om in de praktijk doelgericht te kunnen beoordelen. Daarom zijn er richtlijnen opgesteld voor elk van de vier probleemgebieden. Daarnaast is onderkend dat het beoordelen een beperkt rationeel en een politiek karakter heeft. Dit heeft geleid tot het opstellen van additionele richtlijnen. De complete set richtlijnen luidt:

1. Identificeer partijen aan de hand van standaardrollen in een proces.
2. Operationaliseer het beoordelingsdoel door het als meetdoel met behulp van het measurement goal template te formuleren.
3. Voer versiebeheer van doelen uit.

4. Bepaal winst- en verliespunten in een beoordeling om het verwachtingspatroon van de partijen te managen; de beoordelaar is hierbij moderator.
5. Bepaal de strategie voor het doorlopen van de activiteiten in het beoordelingsproces.
6. Zorg ervoor dat de voor een beoordeling geselecteerde mensen ook daadwerkelijk aan het beoordelingsproces deelnemen (participeren).
7. Onderken expliciet afwegingsmomenten in het proces: de momenten waarop in het proces doel- en middelen worden afgewogen.
8. Expliciteer doel en middelen. Het doel wordt geëxpliciteerd door het te formuleren middels het measurement goal template. De middelen worden geëxpliciteerd door ze te karakteriseren met behulp van attributen. Voorbeelden zijn het object of de kwaliteitskarakteristiek waarop het middel betrekking heeft.
9. Onderken expliciet de momenten waarop er in het proces wordt teruggekoppeld.
10. Meet aan het proces, de activiteiten, bepaal de problemen in het proces en stel de oorzaken van deze problemen vast om bijsturingsinformatie van voldoende 'volwassenheidsniveau' te laten zijn.

Het aldus opgestelde ontwerp is in hoofdstuk 7 geëvalueerd door naar logische consistentie en toepasbaarheid van en naar tevredenheid over het ontwerp te kijken. Hieronder komen resultaten van deze evaluatie aan de orde.

Doelformulering – de doelformulering heeft betrekking op de eerste twee activiteiten uit de processtructuur, namelijk 'bepalen organisatiedoel' en 'bepalen meetdoel'. Verder zijn de eerste vier richtlijnen van belang. De eerste richtlijn betreft het bepalen van partijen aan de hand van de (standaard) rollen in een beoordelingsproces. Deze richtlijn gaat in op een deel van het probleem rondom de doelformulering, namelijk dat doelen te vaag zijn geformuleerd. Door belanghebbenden expliciet te identificeren, wordt het makkelijker om doelen, problemen en meningen in een beoordeling te herkennen en te plaatsen. Tijdens uitgevoerde opdrachten in de praktijk bleek het goed mogelijk om rollen aan belanghebbenden in een beoordeling toe te kennen.

De tweede richtlijn is 'operationaliseer beoordelingsdoel door het als meetdoel met behulp van het measurement goal template te formuleren'. Deze richtlijn heeft betrekking op de activiteit 'bepalen meetdoelen'. We richten ons hiermee op het probleem dat doelstelingen moeilijk in termen van meten zijn te operationaliseren. In de beoordelingspraktijk kon het measurement goal template goed worden toegepast. Ook zijn betrokkenen tevreden over de bereikte formuleringen: het levert meer inzicht op dan de tot dan toe gebruikte doelformuleringen.

De derde richtlijn luidt 'voer versiebeheer van doelen uit'. Deze richtlijn betreft het probleem dat doelen veranderen tijdens een beoordeling. Voor de besturing van het proces is het van belang om te weten welke doelen actueel zijn en op welke de uitwerking van de beoordeling

is gebaseerd. Dit vereist het identificeren van doelen door er versies aan toe te kennen en deze versies te beheren.

De vierde richtlijn is het 'bepalen van de winst- en verliespunten rondom een beoordeling'. Deze richtlijn gaat in op het politieke karakter van beoordelingen. Het verlangt actieve deelname van de beoordelaar als moderator om zo tijdens het proces het verwachtingenpatroon van de verschillende belanghebbenden te managen en een reëel beeld van de beoordeling te schetsen.

Strategie en aansturing – het tweede ontwerpaspect betreft de problematiek rondom aansturing van het proces. Hiervoor is een nieuwe processtructuur ontwikkeld. De richtlijnen 5 en 6 hebben direct betrekking op de aansturing.

Richtlijn 5 betreft het vaststellen van de volgorde van beoordelingsactiviteiten. Hieraan moet richting worden gegeven door het kiezen van een strategie. In het onderzoek zijn drie strategieën onderkend, namelijk: lineair iteratief, inherent iteratief en evolutionair. In de praktijkopdrachten kwamen we de lineaire iteratieve en de evolutionaire strategie tegen.

Richtlijn 6 betreft het idee dat de voor de uitvoering geselecteerde mensen bij het proces betrokken moeten zijn. Om deze participatie te beïnvloeden zijn vier factoren onderkend namelijk: management commitment, motivatie, gebruik van tools en aanbieden van referenties. Ervaringen in de beoordelingspraktijk laten zien dat alle van invloed waren op de participatie in het proces.

Afwegen van doel en middelen – het derde ontwerpaspect betreft de afwegingsproblematiek. De richtlijnen 7 en 8 raken aan deze problematiek. Richtlijn 7 betreft het onderkennen van afwegingsmomenten in het proces waarop er moet worden nagedacht over de inzet van middelen in relatie tot het gestelde doel. Hiervoor zijn in de processtructuur expliciet momenten onderkend.

Richtlijn 8 betreft het idee dat er voor de feitelijke afweging meer informatie over de kwaliteit en kwantiteit van de middelen dient te zijn. Hierbij is een onderscheid aangebracht tussen de benodigde en de aanwezige middelen. De benodigde middelen worden afgeleid uit het meetdoel van de beoordeling. Ze worden geïdentificeerd aan de hand van elementen uit het measurement goal template, namelijk: object, kwaliteitskarakteristiek en perspectief. Een vierde attribuut betreft de kwantiteit van het middel, dit is geformuleerd als inzet en tijd.

De in een beoordeling beschikbare middelen karakteriseren we op vergelijkbare wijze. Wat de in te zetten middelen zijn, verschilt echter per activiteit. Er is dan ook niet een algemeen karakterisering voor middelen te geven. Het onderzoek heeft zich met name gericht op één soort middelen, namelijk: metrieken. Deze identificeren we met behulp van de attributen:

object, kwaliteitskarakteristiek, perspectief en tijd. Daarnaast zal informatie over de validiteit en reproduceerbaarheid van iedere metriek moeten worden bijgehouden. Op basis van deze indeling van attributen is een metriekendatabase ontworpen. Het doel hiervan is het verzamelen van gegevens over toegepaste metrieken, om deze gegevens aan te wenden in nieuwe beoordelingen. De ervaringen met deze metriekendatabase in de praktijk laten zien dat er een begin is gemaakt, maar dat het instrument nog niet voldoet om het afwegingsproces te kunnen ondersteunen. Hieruit is onder andere geconcludeerd dat er ook informatie over de relaties tussen vraag en metrieken moet worden bijgehouden.

Bijsturing en terugkoppeling – het derde ontwerpaspect betreft de problematiek rondom terugkoppeling en bijsturing. De richtlijnen 9 en 10 zijn bedoeld om deze problematiek te addresseren. Richtlijn 9 onderkent dat er expliciet in een processtructuur momenten moeten worden gedefinieerd waarop er wordt teruggekoppeld. Hiervoor zijn drie momenten aan het begin van het proces aangegeven.

Richtlijn 10 betreft het volwasenheidsniveau van de informatie die tijdens terugkoppeling wordt overgedragen. In het onderzoek is aangegeven dat er aan het proces moet worden gemeten, dat de problemen in het proces moeten worden bepaald en dat de oorzaken van de problemen moeten worden geïdentificeerd. Eventueel wordt er voor een volgende keer geleerd over het proces. Dit alles is gericht op het bijsturen van het proces. Onze ervaringen met terugkoppelingsinformatie laten zien dat het voor het identificeren van problemen en oorzaken in een beoordelingsproces belangrijk is dat het proces gestructureerd is uitgevoerd. Selecteren van metrieken, om daarna pas de rationale ervan te bepalen, maakt terugkoppeling zinloos.

Bruikbaarheid van het ontwerp – ons ontwerp betreft een doelgerichte manier voor het beoordelen van softwareproducten. Delen van het ontwerp zijn uitgeprobeerd tijdens drie praktijkopdrachten en bleken goed toepasbaar. Omdat het uiteenlopende beoordelingsbenaderingen betrof –twee codemetrieken gebaseerde benaderingen en één vraaggebaseerde benadering– verwachten we dat ons ontwerp generiek toepasbaar is.

8.2 Aanbevelingen

In dit onderzoek is ingegaan op beoordelingsprocessen en de besturing ervan. De problematiek rondom de besturing is in kaart gebracht en er is een ontwerp opgesteld om dergelijke processen beter te besturen. Naast de oorspronkelijke onderzoeksvragen zijn er tijdens het onderzoek nieuwe vragen opgeworpen. Deze paragraaf besluit daarom met aanbevelingen voor verder onderzoek.

Toetsing – in dit onderzoek zijn besturingsconcepten voor het verbeteren van de besturing van beoordelingsprocessen ontwikkeld. Dit ontwerp is wel geëvalueerd, maar niet echt

getoetst. De eerste aanbeveling is dan ook om het ontwerp beter te toetsen en de vraag te beantwoorden of alle concepten uit het ontwerp een echte verbetering opleveren?

Product- én proces beoordeling – dit onderzoek is gestart vanuit het thema 'beoordelen van softwareproducten'. Het verdient aanbeveling om nader te bekijken in hoeverre procesbeoordelingen –in hoofdstuk 1 aangeduid als assessments– van onze ideeën over productbeoordelingen kunnen profiteren.

Inventariseren van kennis over metrieken – in het ontwerp is aandacht besteed aan het expliciteren van kennis over middelen. Dit is in het proefschrift uitgewerkt door een prototype van een metriekendatabase te ontwikkelen. In hoofdstuk 7 hebben we gezien dat dit prototype (nog) niet voldeed: het kapitaliseren van kennis over metrieken is niet eenvoudig uit te voeren. In dat hoofdstuk is voorgesteld om ook kennis over de relatie tussen metrieken en vragen bij te gaan houden. Het verdient aanbeveling om de toepasbaarheid van een ervaringsbank met metrieken verder te onderzoeken. Er kan wat dit betreft worden aangesloten bij diverse initiatieven op dit gebied, waaronder de Werkgroep Software Metrics van het NESMA/NGGO (Nesma, 1999).

Aandacht voor beoordelingsmiddelen anders dan metrieken – tijdens de analyse van bestaande beoordelingsmethoden is duidelijk geworden dat er veel aandacht wordt besteed aan één type middel, namelijk: metrieken. Ook in dit onderzoek is dit type middel voornamelijk aan de orde geweest. Hierdoor bestaat de indruk dat andere middelen er minder toe doen. Het tegendeel is echter waar. Er is bijvoorbeeld ook behoefte aan middelen om productkwaliteit te specificeren. De Space-Ufo methode is hier een voorbeeld van. Maar ook de in hoofdstuk 6 genoemde 'collaborative' tools zouden ondersteuning kunnen bieden. Het verdient aanbeveling om vervolgonderzoek te richten op dergelijke middelen. Speciale aandacht hierbij verdient de inzet van mensen als beoordelingsmiddel.

Ontwikkelen van beoordelingsscenario's – de in dit proefschrift gedefinieerde processtructuur en het onderkennen van het concept strategie om een proces te doorlopen maakt dat het proces op vele manieren kan worden uitgevoerd. Wij hebben in hoofdstuk 7 echter ook geconstateerd dat we nog maar weinig weten over beoordelingsprocessen. Daarom lijkt ons het ontwikkelen van scenario´s van belang. Een scenario beschrijft de keuzes in een beoordelingsproces in relatie tot de omstandigheden –contingentie factoren– rondom de inrichting van een proces. De behoefte aan scenario's wordt concreet bij het certificeren van softwareproducten (O´Duffy, Jakobsen en Punter, 1999). Certificatie vereist een systeem waarin het beoordelingsproces én de te toetsen normen en waarden worden vastgelegd. Dergelijke systemen worden voor specifieke producten, bijvoorbeeld commerial-of-the-shelf (COTS) producten, fail safe systemen, en vanuit specifieke normen, bijvoorbeeld CCIMB (1999) of ISO 12119 (1994), ontwikkeld. Vanuit onze beschrijving van de processtructuur en de richtlijnen voor besturing kan men mogelijk tot scenario's voor

certificatie komen. De literatuur over het situationeel kiezen van informatiesysteem-ontwikkelingsmethoden (Brinkkemper, 1996), (Punter en Lemmen, 1996) biedt ons inziens aanknopingspunten.

Dit onderzoek is gestart vanuit de constatering dat er in de praktijk ontevredenheid bestaat over softwareproductbeoordelingen. Om dit probleem aan te pakken hebben we met name gekeken naar de besturing van beoordelingsprocessen. Hiervoor is een analysekader ontwikkeld, waarna er een ontwerp is opgesteld om die besturing te verbeteren. Wij hopen dat de concepten in dit proefschrift enerzijds een aanzet zijn voor verder onderzoek en dat ze anderzijds worden toegepast in de praktijk om te komen tot betere beoordelingen van softwareproducten.

9. Literatuur

Abowd, G. e.a., *Recommended best industrial practice for software architecture evaluation*, SEI Technical Report CMU/SEI-96-TR-25, 1996.

Abran, A., P.N. Robillard, *Function point analysis: an empirical study of of its measurement processes*, in: IEEE Transactions on Software Engineering, 1996.

Abreu, F.B.e, R. Carapuca*, Candidate metrics for object-oriented software within a Taxonomy Framework*, Journal of Systems and Software, vol. 26, no. 1, pp. 87-96, 1994.

Afotec, *Software maintainability evaluation guide*, Kirtland (USA), Department of the Air Force, 1996.

AG 29 mei 1998, *Automatisering artsenpraktijk vertraagd – eind dit jaar KemaKeur voor nieuwe systemen – software zal voldoen aan functionele eisen*, in Automatisering Gids van 29 mei 1998.

AG 15 januari 1999, *Kritiek op ontwikkeling IT voor de medische wereld*, in: Automatisering Gids van 15 januari 1999.

AG 5 februari 1999, *KEMA en HIS-leveranciers botsen door keuringseisen*, in: Automatisering Gids van 5 februari 1999.

AG 23 april 1999, *Geknoei met elektronische recepten*, in: Automatisering Gids van 23 april 1999.

AG 20 augustus 1999, *Systeemleveranciers luiden noodklok om rijksmaatregel – huisartssoftware niet tijdig aan te passen aan plannen ministerie VWS*, in: Automatisering Gids van 20 augustus 1999.

AG 21 januari 2000, *Huisartsen balen van hun IT-systemen*, in: Automatisering Gids van 21 januari 2000.

AG 28 januari 2000, *Automatisering huisartsen vraagt om ingrijpen*, in: Automatisering Gids van 28 januari 2000.

AG 18 februari 2000, *'Huisartsen hebben bizar veel macht'*, in: Automatisering Gids van 18 februari 2000.

AG 3 maart 2000, *SMS Cendata laat artsen in grote onzekerheid,* in: Automatisering Gids van 3 maart 2000.

AG 23 juni 2000, *SMS Cendata 'schenkt' Topaas aan gebruikers,* in: Automatisering Gids van 23 juni 2000.

AG 14 juli 2000, *Kibbelende gebruikers verlammen HIS-ontwikkeling,* ingezonden brief, in: Automatisering Gids van 14 juli 2000.

AG 15 september 2000, *Euroned laat lastige huisartsen vallen – dramatisch hoogtepunt in conflict met gebruikersgroep*, in Automatisering Gids van 15 september 2000.

Aken, J.E. van, *De bedrijfskunde als ontwerpwetenschap – de regulatieve en de reflectieve cyclus*, in: Bedrijfskunde, jrg. 66, nr. 1, pp. 16-26, 1994.

Andersen, O. en H. Kyster, *Reproducibility of checklists*, in: Bache en Bazzana, 1994.

Azuma, M., *Software products evaluation system: quality models, metrics and processes - international standards and Japanese practice*, in: Journal of Information and Software Technology, vol. 38, pp.145-154, 1996.

Bache, R. en G. Bazzana, *Software metrics for product assessment*, London, McGraw-Hill Book Company, 1994.

Barbacci, M.R. M.H. Klein, C.B. Weinstock, *Principles for evaluating the quality attributes of a software architecture*, SEI Technical Report CMU/SEI-96-036, 1996

Bartelds, J.F. E.M.W.A. Jansen, Th. H. Joostens, *Enqueteren*, Groningen, Wolters-Noordhoff, 1989.

Basili, V.R., D. Weiss*, A methodology for collecting valid software engineering data*, in: IEEE Transactions on Software Engineering, vol SE-10, no.4, 1984.

Basili, V.R., R.W. Selby, D.H. Hutchens, *Experimentation in software engineering*, in: IEEE Transactions on Software Engineering, vol 12, no. 7, pp. 733-743, 1986.

Basili, V.R., H.D. Rombach, *The TAME Project: Towards improvement-oriented software environments*, in: IEEE Transactions on Software Engineering, vol. SE-14, no. 6, pp. 758-773, 1988.

Baumann, J.M., *Survey of U.S. air force organizations having experience with Afotec's subjective questionnaire-based software maintainability evaluation methods*, Report of results, University of Maryland, 1996.

Baumert, J.H., M.S. McWhinney, *Software Measures and the Capability Model*, SEI Technical report, CMU/SEI-92-TR-25, 1992.

Bemelmans, T.M.A., *Bestuurlijke informatiesystemen en automatisering*, Deventer, Kluwer Bedrijfsinformatie, 1998.

Bersoff, E.H., en A.M. Davis, *Impacts of life cycle models on software*, in: Communications of the ACM, vol. 34, no. 8, 1991.

Bevan, N., *Quality and usability: a new framework*, in: van Veenendaal en McMullan, 1997/

Boegh J., H.L. Hausen, D. Welzel*, Guide to software product evaluation - Evaluator's guide*, ISO/IEC/JTC1 7.13.03, Geneve, 1992.

Boegh J., e.a., *A method for software quality, planning, control and evaluation*, in: IEEE Software, March/April, 1999.

Boehm, B. e.a., *Characteristics of software quality*, TRW Series, of Software Technology, Amsterdam, North Holland, 1978.

Boehm, B., *A spiral model of software development and enhancement*, in: IEEE Computer, vol. 21, no.5, pp.120-131, 1988.

Bongers, F.J. en J.L.A. Geurts, *Beslissen doe je niet alleen; over de rol van GDSS in interactieve besluitvorming*, in: IT-Monitor, blz., 4-6, 11 september 1998.

Briand, L., J. Carriere, R. Kazman en J. Wüst , *Compare: A Comprehensive Framework for Architecture Evaluation*, ISERN report 98-28, 1998.

Briand, L., S. Morasca, V. Basili, *An operational process for goal-driven definition of measures*, ISERN report 99-04, 1999a.

Briand, L., J. Wüst, H. Lounis, *Using coupling measurement for impact analysis in OO systems*, in: Proceedings International Conference on Software Maintenance (ISCM), 1999b.

Brinkkemper, S., *Method engineering: engineering of information systems development methods and tools*, in: Journal of Information and Software Technology, pp. 275-280, 1996.

Brombacher, A.C., *MIR: covering non-technical aspects of IEC 61508 reliability certification*, in: Reliability Engineering Systems and Safety, 66, pp. 109-120, 1999.

Brombacher, A.C., *Designing reliable products in a cost and time driven market: a conflict or a challenge*, Intreerede, Technische Universiteit Eindhoven, 18 februari 2000.

Bruin, A. de, *EDP-auditing, wat is het?*, in: EDP-auditor, jrg. 2, nr. 2, blz. 25-37, 1993.

Bruyninckx, B., e.a., *Software development process handbook - C coding guideline*, Schlumberger RPS rapport RAD-S050.004, 1993.

Caliman, P., *Software product quality evaluation and certification: the Q-Seal methodology*, in: Proceedings EuroStar´96, 1996.

Checkland, P., *Systems thinking, systems practice*, John Wiley & Sons, Chicester, 1981.

Chidamber, S., C. Kemerer, *A metrics suite for object oriented design*, IEEE Transactions on Software Engineering, vol. 20, no. 6, pp.476-493, 1994.

CCIMB, *Common criteria for information technology security evaluation*, ISO version 2.1, 1999.

Courtney, R., D. Gustafson, *Shotgun correlations in software measures*, in: Software Engineering Journal, January 1993.

Daughtrey, T., Kimmel, D., S. Levinson, *Using the Datrix source code analyzer to characterize maintainability*, in: Proceedings of the Bell Canada Quality Engineering Workshop, 1990.

Delen, G., D. Rijsenbrij, *Kwaliteitsattributen van automatiseringsprojecten en informatiesystemen*, in: Informatie, jrg. 32., nr.1, blz. 46-55, 1990a.

Delen, G., D. Rijsenbrij, *Het realiseren en meten van productkwaliteit*, Informatie, jrg. 32., nr.11, blz. 858-867, 1990b.

Deligiannis, I, e.a., *How should we present empirical results to other researchers and other practitioners?*, IEEE WESS-report, September, 1999.

DeMarco, T., *Function and disfunction*, presentatie op European Software Control and Metrics conferences (Escom), Kerkrade, 1995.

Deutsch, M., R. Willis, *Software quality engineering*, Englewood Cliffs, Prentice Hall, 1988.

Dromey, G., *Cornering the Chimera*, in: IEEE Software, January, pp.33-43, 1996.

Dumke R., E. Foltin, R. Koeppe, A. Winkler, *Softwarequalität durch Meßtools - Assessment, Messung und instrumentierte ISO 9000*, Vieweg Braunschweig, 1996.

Eisinga, P., J. Trienekens, M. van der Zwan, *Determination of quality characteristics of software products - concepts and casestudie experiences*, in: Proceedings of 1st World Congress for Software Quality, June 20-22, San Francisco (US), 1995.

Ekris, J. van, *The design of a method for identifying quality requirements for software products*, Master's thesis TUE, april 1998.

Emendo, *Cosmos2000 documentatie*, Hoofddorp, Emendo, 1999.

EN 45001, *General criteria for the operation of testing laboratories*, Cenelec/BSI, 1989.

EN 45002, *General criteria for the assessment of testing laboratories*, Cenelec/BSI, 1989.

EN 45011, *General criteria for certification bodies operating product certification*, Cenelec/BSI, 1989.

Erni, K., C. Lewerentz, *Applying design metrics to OO frameworks*, in: Software Metrics Symposium, IEEE Computer Society press, 1996.

Eijnatten, F. van, *Ontwerpgericht onderzoek*, Hand-out Aio cursus Methodologie van onderzoek en ontwerp, Nobo, 1992.

Fagan, M.E., *Advances in software inspections*, in: IEEE Transactions on software engineering, vol. SE-12, no. 7, pp. 744-751, 1986.

Fenton, N., S. Pfleeger, *Software metrics, a rigorous & practical approach*, London, International Thomson Computer Press, 1996.

Florusse, L.B., M.J.F. Wouters, *Ontwerpgericht wetenschappelijk onderzoek in de bedrijfskunde*, in: Bedrijfskunde jrg 63, nr 2., blz. 237-246, 1991.

Friedman, M., J. Voas, *Software assessment - Reliability, Safety, Testability*, London, Wiley and Sons, 1995.

Fusaro, P, K. El Emam, B. Smith, *Evaluating the interrater agreement of process capability ratings*, ISERN report 97-11, 1997.

Garvin, D., *What does product quality really mean*?, in: Sloan Management review, vol.26, no.1, pp. 25-43, 1984.

Geerts, G. En H. Heestermans, *Van Dale woordenboek der Nederlandse taal*, Utrecht, Van Dale Lexicografie, 1984.

Genuchten, M. van, *Towards a software factory*, proefschrift, TU Eindhoven, Eindhoven, 1991.

Geyres, S., *NF Logiciel, affordable certification for all software products*, in: E. van Veenendaal en J. McMullan, 1997.

Gilb, T., D. Graham, *Software inspection*, Workingham, Addison Wesley, 1990.

Haas, R.J. de, C.S.M. Wubbels, *Situationeel projectmanagement bij automatisering*, in: Informatie, jrg. 32, nr. 2, 1990.

Hall, T., N. Fenton, *Implementing effective software metrics programs*, in: IEEE Software, vol. 14, no. 2, pp. 55-65, 1997.

Halstead, M., *Elements of software science*, Elsevier, 1977.

Hausen, H.L., D. Welzel, *Guides to software evaluation*, Arbeitspapiere der GMD 746, April 1993.

Heemstra, F.J., *Hoe duur is programmatuur?*, proefschrift, Technische Universiteit Eindhoven, 1989.

Heemstra, F.J., R.J. Kusters, J.J.M. Trienekens, *Van kwaliteitsbehoeften naar kwaliteitseisen*, Leidschendam, Lansa publishing, 1994.

Heemstra, F.J., *Software management: van de kunst van het programmeren naar het programmeren van de kunst*, inaugurale rede, Heerlen, Open Universiteit, 1994.

Hoogerwerf, A., *Overheidsbeleid - een inleiding in de beleidswetenschap*, Alphen aan den Rijn : Samsom H.D. Tjeenk Willink, 1989.

Humphrey, W., *Managing the software process*, Reading, Addison-Wesley, 1989

Hutjes, J., J. van Buuren, *De gevalstudie: strategie van kwalitatief onderzoek*, Heerlen, Open universiteit, 1996.

ICEK, *Keuring en certificatie van softwareproducten door Stiching ICEK*, Amersfoort, Stichting ICEK, 1997.

ISO 2382-1, *Information technology – Vocabulary – Part 1: Fundamental terms*, Geneve, ISO/IEC, 1993.

ISO 8402, *Quality management and quality assurance vocabulary*, Geneve, ISO/IEC, 1994.

ISO 9126, *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*, Genève, ISO, 1991.

ISO CD 9126, *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*, part I – IV, Genève, ISO, 1991.

ISO 12119, *Information technology – Software packages – Quality requirements and testing*, Genève, ISO/IEC, 1994.

ISO CD 12182, *Categorization of software,* commission draft, Genève, ISO/IEC, 1994.

ISO 12207*, Information technology – Software life cycle processes*, Genève, ISO/IEC, 1995.

ISO 14598-1, *Information technology – Software product evaluation, Part 1 – General overview*, Genève, ISO/IEC, 1999.

ISO 14598-5, Information technology – *Software product evaluation, Part 5: Process for evaluators*, Genève, ISO/IEC, 1998.

ISO FDIS 14598-6, *Information technology – Software product evaluation, Part 6: Documentation of evaluation modules*, Genève, ISO/IEC, intern rapport, 1999.

ISO 61508, *Functional safety of electrical/electronic/programmable electronic safety-related systems*, Genève, ISO/IEC, 1999.

Kafura, D., G.R. Reddy, *The use of software complexity metrics in software maintenance*, in: IEEE Transactions on Software Engineering, vol. 13, no. 3, pp. 335-343, 1987.

Kazman, R., e.a., *Scenario-based analysis of software architecture*, in: IEEE Software, pp. 47-55, november, 1996.

Kazman, R., e.a., *The Architecture Tradeoff Analysis Method*, in: 4th International Conference on Engineering of Complex Computer Systems, augustus 1998.

Kaposi, A., B. Kitchenham, *The architecture of system quality*, Software Engineering Journal, vol. 2, nr. 1, 1987.

Kirakowski, J., M. Corbett, *SUMI: the software usability measurement inventory*, in: Britisch Journal of Educational Technology, vol. 24, nr. 3, 1993.

Kitchenham, B., S. Lawrence Pfleeger, N. Fenton, *Towards a framework for Software Measurement Validation*, IEEE Transactions on Software Engineering, December, 1995a.

Kitchenham, B., e.a., *Experiences with ISO 9126*, in: Proceedings European Software Control and Metrics conferences (Escom), Kerkrade NL, 1995b.

Kitchenham, B., S. Pfleeger, *Software quality: the elusive target*, in: IEEE Software, January, pp.12-21, 1996.

Kocks, C. *Auditing ge-audit*, in: Informatie, jrg., nr. 11, blz. 28-39, 1997.

Koshgoftaar, T.M., E.B. Allen, *Predicting the order of fault-prone modules in legacy software*, in: IEEE Transactions on Software Engineering, 1998.

Kramer, N.J.T.A., J. de Smit, *Systeemdenken*, Leiden, Stenfert Kroese, 1987.

Kroonenberg, H.H. van den, *Methodologie van ontwerpen*, in: De Ingenieur 86 (47), pp. 915-923, 21 november 1974.

Kyster, H., *MicroScope – the evaluation of software product quality*, DQD-501200, Delta, 1995.

Laguë, B., A. April, *Mapping of Datrix software metrics set to ISO 9126 maintainability sub characteristics*, in: Proceedings Software Engineering Standard issues, Montreal (C), 1996.

Lamprecht, W., C. Weber, *Benefits from user-focused measurement based on GQM*, in: Proceedings of FESMA, Antwerp, 1998.

Latum, F. van, e.a., *Adopting GQM-based measurement in an industrial environment*, in: IEEE Software, 1996.

Leeuw, A.C. de, *Systeemleer en organisatiekunde*, Leiden, Stenfert Kroese, 1974.

Leeuw, A.C. de, *Organisaties: management, analyse, ontwerp en verandering: een systeemvisie*, Assen, Van Gorcum, 1980.

Lemmen, K., T. Punter, M. Dicker e.a., *Methodologie van informatiesysteemontwikkeling*, Heerlen, Open Universiteit Nederland, 1993.

Lewerentz, C., F. Simon, *A product metrics tool integrated into a software development environment*, in: Proceedings of FESMA, Antwerp, 1998.

Li, W, S. Henry, *An empirical study of maintenance activities in two object-oriented system*, in: Journal of Software Maintenance, vol 7, pp.131-147, 1995.

Lions, J.L., *Ariane 5 Flight 501 failure*, report of inquiry board, www.esrin.esa.it/htdocs/tide/press/press96/arianerep.html, 1996.

Lloyd's Register, *Software conformity assessment system - procedure CS94*, London, Lloyd's Register, 1994.

Lundeberg, M., G. Goldkuhl, A. Nilsson, *Systeem ontwikkeling volgens ISAC*, Alphen a/d Rijn, Samson, 1982.

Magee, S., L. Tripp, *Guide to software engineering standards and specifications*, Boston (MA), Artech House, 1997.

Maiocchi, M. *Experiences in evaluating software quality in the banking sector*, in: E. van Veenendaal J. McMullan (eds.), 1997.

Markus, L., *Systems in organisations*, Cambridge (MA), Ballinger publishing company, 1984.

Mayrand, J., F. Coallier, *System acquisition based on software product assessment*, in: Proceedings of International Conference on Software Engineering (ICSE) 18, 1996.

McCabe, T., *A software complexity measure*, in: IEEE Transactions on software engineering, SE-2, no. 4, pp. 308-320, 1976.

McMullan, J., *G.P. Software product accreditation scheme*, Dublin, Centre for Software Engineering, 1994.

Mellor, P, *Critique of ISO/IEC 9126*, Centre for software reliability, City university Londen, 1992.

Meijer, B., *Object oriented software construction*, New York, Prentice Hall, 1988.

Miller, K., D. Dunn, *Post implementation evaluation of IS/IT: a survey of UK practice*, in: Proceedings of 4th European conference on the Evaluation of Information Technology (EVIT), pp. 47-56, 1997.

Moonen, H.B., *Kwaliteitsnormen bij EDP-auditing: een kritische beschouwing*, intreerede, Katholieke Universiteit Brabant, 27 september 1991.

NESMA, *Research Software Metrics – project plan*, Werkgroep software metrics van NESMA/NGGO, 1999.

Niessink, F., *Perspectives on improving software maintenance*, proefschrift, Amsterdam, Vrije Universiteit, 2000.

O'Duffy, M., A. Jakobsen, T. Punter, *Scope mark certification of software product*, in: Proceedings European Software Control and Metrics conferences (ESCOM'99), Herstmonceux (GB), 1999.

Oman, P, J. Hagemeister, D. Ash, *A definition and taxonomy for software maintainability*, Technical report University of Idaho 91-08, 1991.

Oman, P., J. Hagemeister, *Metrics for assessing a software system`s maintainability*, in: Proceedings International Conference on Software Maintenance (ISCM), Orlando, November, 1992.

Omi/Spam consortium, *An overview of the SPAM methodology*, intern rapport (R2.4a), Omi/Spam consortium, 1997.

Park, R., *Software size measurement: a framework for counting source statements*, SEI report, CMU/SEI-92-TR-20, 1992.

Park, R, W. Goethert, W. Florac, *Goal-driven software measurement - a guidebook*, SEI report CMU/SEI-96-HB-002, 1996.

Paulk, M., e.a., *The capability maturity model: guidelines for improving the software process*, Reading, Addison-Wesley, 1995.

Pivka, M., V. Potocan, *How can software packages certfication improve software process*, in: : Proceedings of International Conference on Software Quality, Dublin, 1996.

Pol, M., R. Teunissen en E. van Veenendaal, *Testen met TMAP*, 's Hertogenbosch, Tutein Nolthenius, 1995.

Popper, K., *The logic of scientific discovery*, New York, Harper and Row, 1968.

Praat, J. van, H. Suerink, *Inleiding EDP-auditing - kwaliteitscontrole en beveiliging van informatiesystemen*, Deventer, Kluwer Bedrijfswetenschappen, 1992.

Programming Research, *QA/C – the complete static analysis system - technical overview*, Hersham (UK), Programming Research Ltd, 1994.

Punter, T., K. Lemmen, *The MEMA-model: towards a new approach for method engineering*, in: Journal of Information and Technology, vol. 38, pp. 295-305, 1996.

Punter, T., *Using checklists to evaluate software product quality – measurement concepts for qualitative determination of quality characteristics*, in: Proceedings European Software Control and Metrics conferences (ESCOM), Berlin, pp. 143-149, 1997.

Punter, T., *Developing an evaluation module to assess software maintainability*, in: Proceedings of Conference on Empirical Assessment in Software Engineering (EASE), Keele, 1998a.

Punter, T, *Results of casestudy on Evaluation Module Maintainability (KEMA) and Static Analysis Tool Procedure (Schlumberger)*, Space-Ufo Technical Report D3112-B, Annex G, 1998b.

Punter, T., G. Lami, *Factors of software quality evaluation*, in: Proceedings European Software Control and Metrics conferences (ESCOM), pp.257-266, Rome, 1998.

Punter, T., *Do your metrics predict software quality?*, in: Proceedings FESMA, Amsterdam, 1999.

Qiu, F., *An expert system approach to modelling and planning software product assessment and certification*, proefschrift, Glasgow Caledonian University, 1995.

Rae, A., H.L., Hausen, P. Robert, *Software evaluation for certfication - principles, practice and legal liability*, London, McGraw-Hill, 1995.

Reeken, A., J. Trienekens, *Het praktisch belang van methoden en CASE-tools –resultaten van empirisch onderzoek*, in: Informatie, jrg. 33, nr. 7/8, pp. 465 – 552, 1991.

Rees, J. van, *De methode werkt niet!*, in: Informatie, 24, pp. 81-93, 1982.

Reifer, D.J., *Software Management - fourth edition*, Los Alamos, IEEE Computer Society Press, 1993.

Renkema, T., *Investeren in de informatie-infrastructuur – richtlijnen voor besluitvorming in organisaties*, proefschrift, Technische Universiteit Eindhoven, 1996.

Revai, B., *Performing static tests*, Schlumberger RPS report, 1997.

Robert, Ph., *Final report Scope*, Scope consortium, 1994.

Robillard, P.N., D. Coupal, F. Coallier, *Profiling Software Through the Use of Metrics*, in: Proceedings Quality Engineering Workshop, Ottawa, 16-17 October, 1991.

Rocha, R. da, S. Palermo, *Software quality assurance in HEP*, in: Computer Physics Communications 57, pp.524-527, 1989.

Rombach, H.D., V.R. Basili, *Quantitative assessment of maintenance*, in: Proceedings International Conference on Software Maintenance (ISCM), 1987.

Rout, T.P., *Consistency and conflict in terminology in software engineering standards*, report Software Quality Institute, Griffith University (Australia), 1998.

Sahraoui, H.A. and D. Azar, *Quality estimation models optimization using genetic algorithms: case of maintainability*, in: Proceedings of Fesma'99, Amsterdam, 1999.

Schneidewind, N.F., *Integration of Software process and product measurement and models*, presentation at Software Reliability symposium, Eindhoven University of Technology, November 6, 1998.

Scope consortium, *Certification model - final report*, 75 pp., 31 August 1990.

Scope consortium, *A method for software assessment and certification – the informal model,* 18 February 1992a.

Scope consortium, *Catalogue of assessment bricks*, 13 March 1992b.

Scope consortium, *Software quality assessment procedures*, February 1993.

Scope consortium , *Summary of SCOPE second phase casestudies*, 27th May 1993b.

Scope IPSE consortium, *Minutes of 3rd Meeting*, Barcelona, 27 October 1994.

Serc, *Het specificeren van softwarekwaliteit*, Deventer, Kluwer Bedrijfswetenschappen, 1992.

Shaw, M. *Prospects for an engineering discipline of software*, in: IEEE Software, November, pp. 15-24, 1990.

Shepperd, M., M. Cartwright, *An empirical investigation of an object oriented software system*, in: Journal of Information and Software Technology, 1999.

Smith, D.J., J.S. Edge, *Quality procedures for hard- and software*, New York, Elsevier, 1991.

Solingen, R. van, E. Berghout, *The Goal/Question/Metric method – a practical guide for quality improvement of software development*, London, McGraw-Hill, 1999.

Solingen, R. van, *Product focused software process improvement*, proefschrift, Technische Universiteit Eindhoven, 2000.

Space-Ufo consortium, *The Space Ufo Methodology - User guide*, Esprit project P22290, 1998.

Spirits*, Static test tool integration*, Schlumberger RPS report, 1997

Squid consortium, *Workpackage 3 - Squid conceptual handbook*, report D3.7/1, Esprit project P8436, June 1996.

Stark, G.E., P. Oman, *A survey instrument for understanding the complexity of software maintenance*, in: Journal for Software Maintenance: Research and Practice, vol 7, pp. 421-441, 1995.

Steels, L., *Kennissystemen*, Amsterdam, Addison-Wesley, 1992.

Strien, P.J. van, *Praktijk van wetenschap*, Assen, Van Gorcum, 1986.

Symons, C., *Improving the accuracy of software sizing and estimating from the functional requirements: back to 1911*, presentatie op Fesma'98, 1998.

Syperski, C., *Component software - beyond object oriented programming*, London, Addison-Wesley, 1998.

Swede, V. van, J.C. van Vliet, *A flexible framework for contingent information systems modelling*, onderzoeksrapport IR-310, Vrije Universiteit, Amsterdam, 1992.

Ticket, *Guide to software quality management systems - construction and certification using EN 29001*, TickIT Office, London, 1995.

Trienekens, J., *Tijd voor kwaliteit*, proefschrift, Technische Universiteit Eindhoven, 1994.

Tsukumo, A.M., A. Pentenado de Oliviera, e.a., *The Second Experiment of Application of ISO 9126 Standards on Quality Evaluation of Brazilian Software Products*, in: Proceedings 6th IntConfSoftwQuality, Ottawa, 1996.

Uittregt, A., *Product focused software process improvement: integrating SPI and SPQ approaches into a quality improvement method for RPS*, afstudeerverslag, Technische Universiteit Eindhoven, 1998.

Veenendaal, E. van, J. McMullan (eds.), *Achieving software product quality*, Den Bosch, Tutein Nothenius, 1997.

Veld, J. In't, *Analyse van organisatieproblemen*, Leiden, Stenfert Kroese, 1988.

Verhoef, D. en V. van Swede, *Euromethod*, Deventer, Kluwer Bedrijfswetenschappen, 1995.

Verilog, *Software development using Logiscope*, Toulouse, Verilog SA, 1992.

Verilog, *Logiscope C/C++ Audit - Basic concepts*, Toulouse, Verilog SA, 1998.

Verschuren, P., H. Doorewaard, *Het ontwerpen van een onderzoek*, Utrecht, Lemma, 1995

Visser, M, *Referentiemodel bij beoordelen van onderhoudbaarheid van software*, intern rapport Kema Nederland B.V., 1997.

Wassink, R., *Stage eindverslag Haagse hogeschool / KEMA*, Den Haag, Haagse Hogeschool, 1998.

WCIA, *WCIA-HIS-Referentiemodel 1995 – deel A – Functionele eisen*, 1996.

WCIA, *WCIA-HIS-Referentiemodel – deel B - Keuringsplan 1e fase HIS-pakketten*, versie 2.1, 1999.

Webster, M. *(ed.), Webster's comprehensive Dictionary of the English language*, Trident press international, 1996.

Welker, K., P. Oman, G. Atkinson, *Development and application of an automated source code maintainability index*, in: Journal of Software Maintenance, Vol.9, pp 127-159, 1997.

Welzel, D., H.L. Hausen and J. Boegh, *Metric-based software evaluation method*, in: Proceeding of BCS 1st European International conference on Software testing, Analysis and Review, London, October 25-28, 1993.

West, *Improving the maintainability of software*, Norwich, CCTA, 1994.

Wijbrans, K. F. Buve, W. Geurts, *Practical experiences in the BOS-project*, in: Proceedings of Embedded systems conference, Eindhoven, 1999.

Wilkie, F.G., B. Hylands, *Measuring complexity in C++ application software*, in: Software practice and experience, pp.513-546, April 1998.

Willumeit, H, G. Gediga, K Hamborg, *The ISO-metrics usability inventory: an operationalization of ISO 9241-10*, Technical report University of Osnabruck, 1996.

Xenos, M., D. Stavrinoudis, D. Christostodoulakis, *The correlation between developer-oriented and user-oriented software quality measurements*, in: Proceedings of International Conference on Software Quality, Dublin, pp.267-275, 1996.

Xenos, M., D. Christostodoulakis, *Measuring perceived software quality*, in: Information and Software Technology, Vol. 39, pp.417-424, 1997.

Yin, R.K., *Casestudie Research, Design and Methods*, New York, Sage, 1994.

Yourdon, E., *Modern Structured Analysis*, Yourdon press, 1989.

Zwaan, A.H. van der, *Organisatie onderzoek*, Assen, van Gorcum, 1995.

Zwaan, A.H. van der, *Van geval tot geval; ontvouwen of beproeven? Over onderbenutting van gevalstudies*, in: Nobo Methodologie cursus, 1998.

Zwan, M. van der, *Een kwaliteitsprofiel – de basis voor het keuren van software product kwaliteit,* afstudeerverslag, Technische Universiteit Eindhoven, 1995.

Zwan, M. van der, T. Punter, *Interpretation of engineer opinion,* intern rapport Océ Research and Development, Océ Technologies, November 1999.

Zuse, H., *Software complexity: measures and methods*, Berlin, de Gruyter, 1990.

Zuse, H., *A framework of software measurement*, Berlin, de Gruyter, 1998.

10. Summary

During the last decade much has been achieved in the area of software product evaluation. The ISO 9126 standard has provided a vocabulary to describe software quality. The ISO 14598 standard has defined the evaluation process. Furthermore many metrics, checklists and other types of tools have been defined to support this process. Much of this has resulted from Esprit projects, especially Scope and Space-Ufo.

Despite this progress, we have noticed that in evaluation practice there is dissatisfaction about software product evaluations. Sometimes an evaluation provides an answer to the wrong question. In such evaluations appropriate activities and up-to-date tools might have been used, but there is insufficient management of these activities, leading to insufficient matching of goals and resources and insufficient feedback during the process to check whether the targeted goal has been achieved. To put it briefly: software product evaluations are often insufficiently controlled. This problem statement was the starting point for the research presented in this thesis.

For this research the problems concerning control are analysed and a design to improve the control is set up. This has resulted into two conceptual models. The first conceptual model is the analysis framework to characterise software product evaluations. It has been described in chapters 3, 4 and 5. The second conceptual model concerns the design to control evaluation processes (chapters 6 and 7). This design is denoted as a goal oriented evaluation. This approach is meant to improve evaluation process control such that these processes deliver the results that are required by the organisation or individual who gave the instructions for the evaluation.

Framework for analysis

The construction of the analysis framework starts with the assumption that evaluation is considered as a process: a set of activities. Several descriptions of such processes can be found in the literature. For example, each evaluation method chooses its own description of evaluation activities and uses its own sequence of these activities. A standard process description is provided by ISO 14598, which was taken as the starting point for this research.

Apart from a process perspective this research has taken a control perspective. This perspective is elaborated by using system theory. This has resulted in a framework for analysis that consists of five aspects, namely:

1. Activities – description of activities: what is happening during the process?
2. Process structure – the coherence between the activities. This provides the sequences of the activities.

3. Management of activities – this is about organising the process (which activities are necessary to achieve the goal), the allocation of resources to the process and recognising that progress control is necessary.
4. Balancing goals and resources – this concerns the deliberation about allocating the resources to the process. Important for this are the quality and quantity of the resources.
5. Feedback – an evaluation process should also be managed when original goals cannot be achieved. This requires feedback: it should be noticed that something goes wrong in the process.

This framework is used to analyse whether existing evaluation methods pay attention to these aspects of control. This is conducted by doing a literature study. The analysis shows that existing evaluation methods pay insufficient attention to the last three aspects: management of activities, balancing goals and resources, and feedback. During the literature study it emerged that goal definition is an important element of control. However this is often neglected by existing methods.

This highlighted the problem statement: results of evaluation of software products are often insufficient due to a lack of well-founded control of the process. Shortcomings of the control are in the following areas:

- Insufficient attention to formulate operational goals, in a way that the involved parties recognise the goal(s).
- Insufficient insight into the activities of the process and the coherence between the activities. Therefore there is a lack of management.
- Insufficient attention for the balancing of goals and resources. Existing evaluation methods hardly recognise this issue. Instead they presume fixed goals and infinite capacity of means.
- Insufficient attention for the feedback of results, which might be followed by adjusting goals, resources and activities.

This problem statement was tested during two case studies. In both cases the framework for analysis enabled identification of the control problems. This confirms the relevance of the four problem areas. Therefore they were used as a starting point for a design to improve evaluation control.

Design

The design aims at goal oriented evaluation of software products. This means that the goals are the starting point of the evaluation. It consists of a process structure and a set of guidelines. The process structure is a set of activities and relations between these activities. It is an adjustment of the structure defined by the ISO 14598 standard. The new process structure recognises relationships between goals and resources explicitly. The

GoalQuestionMetric-principle is used for the implementation. The whole process consists of nine activities, namely:

- Determine organisation goal.
- Determine measurement goal.
- Define questions and hypotheses.
- Select metrics and determine thresholds.
- Determine criteria model.
- Formulate evaluation plan.
- Conduct measurement.
- Compare to criteria.
- Determine evaluation results.

The process structure is not sufficient to achieve goal-oriented evaluation in practice. Therefore guidelines have been defined for each of the four problem areas. Besides it has been recognised that any evaluation process will not be entirely rational but will have a political aspect. This has resulted into a set of additional guidelines. The complete set of guidelines is presented below:

1. Identify parties by standard roles in the evaluation process.
2. Make the evaluation goal operational by formulating it as a measurement goal; make use of the measurement goal template.
3. Conduct version management on the evaluation goals.
4. Determine the profit and loss of an evaluation to manage the expectations of these parties. An independent evaluator might play a mediating role here.
5. Determine a strategy to guide the sequence of process activities.
6. Ensure that the selected people actually participate in the evaluation process.
7. Determine explicitly moments in the process to balance goals and resources.
8. Describe goals and resources explicitly. Goals are made more explicit by formulating them using a measurement goal template. The resources are made explicit by characterising them by product attributes.
9. Determine explicitly moments in the process to conduct feedback.
10. Measure the process, determine the problems in the process and determine the causes of these problems to obtain mature management information.

The design was evaluated on three levels: logical consistency, applicability and user satisfaction. Logical consistency is discussed in chapter 6 where the design was constructed. Chapter 7 deals with applicability and user satisfaction. To evaluate the design on these two levels three evaluation assignments were conducted. These cases confirmed the finding of this thesis.

11. Index

12. Curriculum vitae

Teade Punter werd geboren op 4 december 1965 te Oosterwolde (Friesland). Hij behaalde zijn Vwo-diploma in 1985 aan het Ichthus college te Drachten. Vervolgens studeerde hij aan de Technische Universiteit Twente. Eerst behaalde hij zijn propedeuse in de Technische Bedrijfskunde. Daarna studeerde hij Wijsbegeerte van Wetenschap, Technologie en Samenleving. Deze ingenieursopleiding werd in 1991 afgerond met een studie naar het bepalen van de toegevoegde waarde van investeringen in informatietechnologie bij de provincie Overijssel.

In 1992 trad Teade in dienst als cursusteamleider bij de Open Universiteit Nederland te Heerlen. Eerst was hij mede verantwoordelijk voor de ontwikkeling van de cursus 'Methodologie van Informatiesysteem ontwikkeling'. Vervolgens was hij cursusteamleider voor de opleidingsvariant Bedrijfskunde binnen de Informatica opleiding. Hij was hier verantwoordelijk voor afstudeerbegeleiding en ontwikkelde ook diverse cursussen. Door dit werk werd zijn interesse voor software management en met name software-kwaliteit gewekt. Ook ontstond de behoefte om zich voor langere tijd met één onderwerp bezig te houden.

Daarom maakte hij in 1996 de overstap naar de Technische Universiteit Eindhoven om zo promotieonderzoek te kunnen doen. Tijdens dit onderzoek op het gebied van het beoordelen van software-kwaliteit maakte hij deel uit van het Software Product Quality-team van Kema Nederland b.v. te Arnhem. Het onderzoek leidde tot diverse publicaties op internationale conferenties en resulteerde in het voorliggende proefschrift.

Sinds 1 oktober 2000 werkt Teade bij het Fraunhofer Institut Experimentelles Software Engineering (IESE) te Kaiserslautern. Hij houdt zich daar als groepsleider bezig met software proces assessments, meetprogramma's en softwareproduct-beoordelingen.