

COMPACT

TIJDSCHRIFT EDP-AUDITING

ONTWIKKELOMGEVINGEN

1997 / 6

INHOUDSOPGAVE

Compact ©

Jaargang 24, nummer 6
Een uitgave van KPMG EDP
Auditors NV en Sansom Bedrijfs-
Informatie, werkmanschap van
Wolters Kluwer NV.

Het blad verschijnt 6 x per jaar.

Redactie

Prof. A.W. Neisingh RE RA

(hoofredacteur)

J.C. Boer RE RA

Ir. J.A.M. Donkers RE

Drs. R.G.A. Fijneman RE RA

J.C. van Praat RE RA

Ir.dr.s. J. van der Vlugt

Adviesraad

Mv. P. van Dijken

G. van Essen RA

Prof.mr. H. Franken

Dr. K.I.J. Moilenna RA

Prof. H.B. Moonen RE RA

Prof.dr.ir. R. Paans RE

Redactiesecretariaat

Mvo. I. de Koning,

Sansom Bedrijfsinformatie,

Postbus 4,

2400 MA Alphen aan den Rijn

Tel.: 0172 - 466 746

Fax : 0172 - 466 569

Vormgeving

Bureau Karakter, Delft

Opmaak

AlphaZet bv, Waddinxveen

Abonnementen

f 165,- per jaar incl. BTW.

Losse nummers f 45,- incl. BTW.

Studentenabonnement f 95,-

incl. BTW. Abonnementen kunnen

schriftelijk tot uiterlijk één maand

voor de aanvang van een nieuw

abonnementsjaar worden opgezegd.

Bij niet tijdige opzegging wordt het

abonnement automatisch met een

jaar verlengd.

Abonnementsadministratie

Sansom Bedrijfsinformatie,

Postbus 4,

2400 MA Alphen aan den Rijn

Tel.: 0172 - 466 800

Fax : 0172 - 475 933

Adreswijzigingen - ook tijdelijke -

moeten minstens 8 weken voor de

verschijningsdatum bekend zijn.

Overname artikelen

Het overnemen en vermenigvuldigen

van artikelen en berichten is

slechts geoorloofd na schriftelijke

toestemming van de uitgever.

Overdrukken artikelen

Overdrukken van artikelen kunnen

worden aangevraagd bij het

redactiesecretariaat. Prijs per over-

druk per artikel (inclusief omslag)

f 5,-.

Uitgever

Dr. J.H. Elich

Uitgeversverbond
Groep vaktijdschriften

Lid van de Nederlandse organisatie
van tijdschriftuitgevers NOTU

ISSN 0920 - 1645

3

Softwaremetingen binnen het Capability Maturity Model

Drs. C.M. Piek

Softwareorganisaties hebben een grote behoefte om hun softwareprocessen te verbeteren. Organisaties kunnen baat hebben bij een raamwerk waarbinnen deze softwareprocessen meetbaar kunnen worden gemaakt zodat inzicht wordt verkregen om verbeteringen te kunnen doorvoeren. In dit artikel wordt een methode gepresenteerd waarmee binnen het CMM een willekeurig softwareproces kan worden gemeten. Daarbij wordt de methode toegepast op het correctieve onderhoudsproces van een softwareleverancier.

15

Selectie van CASE-hulpmiddelen: een casebeschrijving

Drs. ing. R.F. Koorn CISA en J.C.J.M. Vrakking

De automatiseringsondersteuning van de systeemontwikkeling evolueert van afzonderlijke tools naar geïntegreerde CASE-omgevingen. Interne ontwikkelafdelingen worden gedwongen applicaties efficiënter en sneller te realiseren, onder druk van de marktontwikkelingen, de gebruikersorganisatie en de dreiging van uitbesteding. Het selecteren en invoeren van een CASE-hulpmiddel is een ingrijpend en kostbaar project. In dit artikel worden de belangrijkste keuzecriteria en succes- en faalfactoren aan de hand van een praktijksituatie stapsgewijs behandeld.

27

Benchmarking van systeemontwikkeling

Drs. J.C. de Boer, ir. J.A.M. Donkers RE en ir. K.M. Lof

Met het toenemende belang van informatietechnologie voor organisaties worden er steeds hogere eisen gesteld aan de kwaliteit van informatiesystemen en de wijze waarop deze binnen de systeemontwikkelingsorganisatie totstandkomen. Een belangrijk element van de besluitvorming over de noodzakelijke kwaliteitsverbeteringen is de vraag hoe de concurrenten het systeemontwikkelingsproces beheersen. Dit artikel beschrijft de uitkomsten van benchmarkonderzoeken die als doel hadden een antwoord te geven op deze vraag.

33

Multimedia in open omgevingen; de mogelijkheden van het World Wide Web

Drs. A.M. Buren

In toenemende mate worden bedrijfssystemen door websites gekoppeld aan het World Wide Web. De ontwikkeling van web-toepassingen stelt specifieke eisen aan de ontwikkelmethode, het ontwikkelproces en de ontwikkelhulpmiddelen. In dit artikel worden de mogelijkheden, ontwikkelingen en auditaspecten besproken die bepalend zijn voor het succes van dergelijke toepassingen.

REDACTIONEEL

Het blad wil een bijdrage leveren aan de ontwikkeling van het vakgebied EDP-auditing door het publiceren van actuele artikelen op deelterreinen van EDP-auditing en adviezen, zoals:

- beoordeling automatiseringsorganisaties en -systemen
- risicobeheersing
- telecommunicatie-adviezen
- beveiligingsonderzoeken
- quality assurance
- opleidingen en trainingen
- privacywetgeving
- computercriminaliteit en nieuwe regelgeving.

Behalve voor EDP-auditors kan dit blad ook interessant zijn voor IT-deskundigen en gebruikers van informatiesystemen. De in dit tijdschrift weergegeven meningen mogen niet worden gezien als officiële zienswijze van KPMG EDP Auditors NV.

Het blad Compact is met de meeste zorg samengesteld. Niettemin is het niet geheel uitgesloten dat de geboden informatie enkel en alleen door tijdsverloop en/of andere oorzaken minder juist is. Noch KPMG, KPMG EDP Auditors, noch de redacteurs persoonlijk, noch uitgeverij Samsom Bedrijfsinformatie BV, deel uitmakend van Wolters Kluwer NV, aanvaarden enige aansprakelijkheid, hoe ook genaamd, uit welken hoofde dan ook voor enig gevolg rechtstreeks of indirect voortvloeiend uit het gebruik van de informatie.

De redactie stelt gaarne ruimte in Compact beschikbaar voor reacties en/of ervaringen van lezers. Auteurs die overwegen een bijdrage te leveren, wordt verzocht kennis te nemen van de aanwijzing voor auteurs, die bij het secretariaat verkrijgbaar is.

De afgelopen jaren is bij de ontwikkeling van nieuwe systemen in toenemende mate de aandacht verschoven van de ontwikkelomgeving naar veranderingen in de bedrijfsprocessen. Belangrijke redenen hiervoor zijn de ogenschijnlijke stabilisering van ontwikkelomgevingen en de enorme rendementen die worden voorgespiegeld bij het uitvoeren van business (process) redesign-trajecten. Ontwikkelomgevingen stonden dan ook steeds minder in de belangstelling van het management.

Op dit moment staan ontwikkelomgevingen weer meer in de aandacht. Noodzakelijke aanpassingen van onder meer programmatuur als gevolg van de millenniumproblematiek hebben hier een belangrijke bijdrage aan geleverd. Naast deze 'negatieve' aandacht voor ontwikkelomgevingen zijn er vanuit de snel veranderende bedrijfsprocessen ook meer 'positieve' redenen om aandacht te besteden aan deze omgevingen. Moderne ontwikkelomgevingen verbeteren immers de flexibiliteit en aanpasbaarheid van informatiesystemen. Hierdoor worden de snelle (organisatorische) veranderingen die het management vaak voorstaat, mogelijk gemaakt.

Om verbeteringen in de ontwikkelomgeving te kunnen doorvoeren is het van belang de bestaande situatie meetbaar te maken. Daardoor is er behoefte aan raamwerken voor de meetbaarheid van softwareprocessen. In het eerste artikel wordt ingegaan op deze problematiek op basis van het Capability Maturity Model. Daarbij wordt de methode toegepast op het correctieve onderhoudsproces van een softwareleverancier.

CASE-hulpmiddelen spelen een belangrijke rol bij het efficiënter en sneller kunnen realiseren van applicaties. Echter, het selecteren en invoeren van een CASE-hulpmiddel is een ingrijpend en kostbaar project. Het tweede artikel behandelt de belangrijkste keuzecriteria en succes- en faalfactoren aan de hand van een praktijksituatie.

In het derde artikel wordt een beschrijving gegeven van de uitkomsten van benchmarkonderzoeken op het gebied van systeemontwikkeling. Gelet op het belang van systeemontwikkeling is de door de auteurs getrokken conclusie, dat 52% van de organisaties geen of slechts informele beheersmaatregelen heeft genomen ten aanzien van systeemontwikkeling, een signaal dat te denken geeft.

Nieuwe soorten van systemen, bijvoorbeeld bedrijfssystemen door websites gekoppeld aan het World Wide Web, hebben grote invloed op ontwikkelomgevingen. In het laatste artikel wordt ingegaan op mogelijkheden, ontwikkelingen en auditaspecten die bepalend zijn voor het succes van dergelijke toepassingen.

Al met al komt een breed scala van onderwerpen aan de orde die alle de bestudering en overweging waard zijn. Volgens de redactie bieden de onderwerpen voor EDP-auditors een goede mogelijkheid om hun rol en houding ten aanzien van de beheersing van systeemontwikkelomgevingen te beschouwen.

De redactie wenst de lezer veel plezier en succes met het toepassen van de vrucht van deze beschouwing in zijn of haar praktijk.

Ir. J.A.M. Donkers RE

Softwaremetingen binnen het Capability Maturity Model

Drs. C.M. Piek

Het Capability Maturity Model (CMM) is door de software-industrie als kader algemeen geaccepteerd om de kwaliteit van haar softwareprocessen (en in het verlengde hiervan haar softwareproducten) continu te kunnen verbeteren. Deze verbeteringen dienen echter gepaard te gaan met softwaremetingen. Hiervoor is een methode nodig die binnen het CMM gebruikt kan worden om de geleverde 'kwaliteit' van een softwareproces te meten.

INLEIDING

De jaren negentig laten zich kenmerken door de bewustwording in de software-industrie dat continue kwaliteitsverbetering van doorslaggevende betekenis kan zijn om op de langere termijn te kunnen overleven. Dit kan onder andere verklaard worden doordat de 'markt' steeds hogere kwaliteitseisen stelt aan softwareproducten.

Voor een blijvend betrouwbare werking van de informatievoorziening dienen onder andere softwareproducten van hoge kwaliteit te zijn. Daardoor worden softwareorganisaties gedwongen om kwaliteitssystemen te ontwerpen en in te richten die kwalitatief hoogwaardige softwareproducten kunnen voortbrengen. Een kwaliteitssysteem dat ontworpen is voor softwareorganisaties is het Capability Maturity Model van Humphrey ([Hump89]).

Het CMM wordt dan ook door softwareorganisaties gebruikt om verbeteringen in hun softwareprocessen te kunnen aanbrengen. Het merendeel van deze organisaties bevindt zich momenteel op CMM level 1 en er worden diverse acties ondernomen om dit niveau te ontstijgen ([Sass96]). Om kwaliteitsverbeteringen te kunnen aanbrengen, dient de organisatie kwaliteitskringlopen te doorlopen om uiteindelijk op een hoger gelegen CMM-niveau te kunnen komen ([Demi82], [Hump89]).

Tijdens het incrementeel doorlopen van deze kringlopen kunnen softwaremetingen en softwarekwaliteitsmetriekeken een organisatie helpen. Om verbeteringen in de ontwikkeling en het onderhoud van softwareproducten te kunnen realiseren, is het voor een organisatie noodzakelijk inzicht te verkrijgen in die factoren die de softwarekosten en -activiteiten beïnvloeden. Software(kwaliteits)metriekeken kunnen dit inzicht geven. Door het definiëren en analyseren van metriekeken en bijbehorende metingen die betrekking hebben op softwareactiviteiten kan een organisatie inzicht krijgen om verbeteringen door te voeren in zowel de kwaliteit van het softwareproces als het softwareproduct zelf ([Grad87], [Neil90]).

Echter, organisaties ondervinden problemen bij het toepassen van (bestaande) softwaremeetmethoden met betrekking tot het definiëren en implementeren van een stelsel van metriekeken die specifiek zijn afgestemd op de karakteristieken van de softwareontwikkelings/onderhoudsomgeving en de behoeften van de desbetreffende organisatie ([Roch94]).

Naast de methode die in dit artikel zal worden gepresenteerd, waarmee binnen het CMM een willekeurig softwareproces kan worden gemeten, zal de methode worden toegepast op het correctieve onderhoudsproces van een softwareleverancier.

CAPABILITY MATURITY MODEL

Het CMM maakt onderscheid tussen een vijftal niveaus van volwassenheid waarop een softwareorganisatie zich kan bevinden: Initial, Repeatable, Defined, Managed of Optimizing. Het onderscheid dat in dit model tussen de verschillende niveaus wordt aangebracht, berust op het idee dat het ene softwareproces volwassener is dan een ander softwareproces. Tevens wordt op basis van ervaringskennis uiteengezet hoe een softwareorganisatie van het ene niveau naar een hoger gelegen niveau kan 'groeien'.

CMM-fundamenten

Voor een beter begrip van het CMM dienen ten eerste een aantal definities te worden gegeven ten aanzien van softwareprocessen ([Paul93]):

- *Software proces capability* is de mate waarin bepaalde voorgedefinieerde uitvoer kan worden verwacht wanneer het desbetreffende softwareproces wordt uitgevoerd.
- *Software process performance* is de prestatie waarmee processen bepaalde uitvoer voortbrengen. Derhalve richt de *process performance* zich op de prestaties zoals die door het desbetreffende proces worden geleverd, terwijl de *process capability* zich richt op de resultaten dan wel de uitvoer die van dit proces mag worden verwacht.
- *Software process maturity* is de mate waarin het softwareproces kan worden gedefinieerd, gestuurd, gemeten en gecontroleerd. Hierbij dient te worden opgemerkt dat naarmate de *volwassenheid* van het softwareproces toeneemt eveneens de *process capability* en de *process performance* zullen toenemen.

CMM: vijf niveaus van volwassenheid

Zoals gesteld kent het CMM vijf niveaus van volwassenheid.

Niveau 1: Initial software process

Het eerste niveau, het *initial* proces, wordt gekarakteriseerd door een ad-hocbenaderingswijze van het softwareproces. De eisen waaraan de invoer van het proces moet voldoen, is niet of slecht gedefinieerd. Tevens wordt er een bepaalde uitvoer verwacht en heeft de organisatie geen inzicht in de transformatie van de invoer naar de uitvoer. Hierom worden de softwareprocessen op dit niveau *initial* genoemd.

Niveau 2: Repeatable software process

Door de in het verleden opgedane ervaringen in het ontwikkelen en onderhouden van min of meer dezelfde softwareproducten heeft de organisatie een zekere statistische controle over kosten en tijdschema's met betrekking tot het ontwikkelen van softwareproducten verkregen. Hierom worden de softwareprocessen op dit niveau *repeatable* genoemd.

De introductie van nieuwe softwaretools, de ontwikkeling van een nieuw type softwareproduct en belangrijke organisatorische veranderingen behoren

tot de belangrijkste risicogebieden op dit niveau. De hoofdactiviteiten om dit niveau te ontstijgen (naar CMM-niveau 3) zijn:

- oprichten van een softwareprocesgroep. Een kleine eenheid die het verbeteren van het softwareproces als enige (primaire) taak heeft;
- beschrijven van een softwareprocesmodel. Met behulp van het in de paragraaf 'Softwareprocesmodel' beschreven procesmodel kan een raamwerk, waarbinnen specifieke ontwikkelings- en onderhoudsprocessen zijn gedefinieerd, in meetbare termen worden gemodelleerd;
- de introductie van een verzameling softwaremethoden en -technieken, zoals te gebruiken standaarden, ontwerp- en testmethoden en programmacode-inspectietechnieken.

Niveau 3: Defined software process

Op CMM-niveau 3 zijn de activiteiten gedefinieerd waaruit het softwareproces is opgebouwd. Tevens zijn de eisen gedefinieerd waaraan de invoer en de uitvoer van deze deelprocessen moeten voldoen. Hierom worden de softwareprocessen op dit niveau *defined* genoemd.

Op dit niveau heeft de organisatie een basis bereikt om continu verbeteringen door te kunnen voeren. Door het beoordelen en analyseren van de bestaande softwareprocessen kunnen probleemgebieden worden geïdentificeerd om verbeteringen te kunnen aanbrengen. De hoofdactiviteiten die op dit niveau onderscheiden kunnen worden, zijn:

- definiëren van een verzameling softwaremetrieken en het ontwerpen van een database om de gegevens die tijdens het softwareproces worden vastgelegd, te kunnen verzamelen en te onderhouden;
- opstellen van een softwarekwaliteitsplan waarin kwaliteitsdoelen zijn geformuleerd. De mate waarin deze doelstellingen zijn gerealiseerd, wordt beoordeeld door kwantitatieve metingen uit te voeren.

Niveau 4: Managed software process

Organisaties hebben op dit niveau kwantificeerbare kwaliteitsdoelstellingen geformuleerd voor zowel hun softwareproducten als -processen. In een softwaremeetprogramma wordt beschreven hoe de doelstellingen dienen te worden gemeten. In een organisatiebrede database worden de gegevens vastgelegd om de desbetreffende metingen te kunnen uitvoeren. Deze metingen vormen een kwantitatieve basis om softwareprocessen en -producten te kunnen evalueren. Tevens wordt een organisatie in staat gesteld trends in de kwaliteit van haar softwareproducten en -processen te kunnen voorspellen binnen voor de organisatie acceptabele grenzen.

Niveau 5: Optimizing software process

Op dit niveau heeft de softwareorganisatie continu aandacht voor softwareprocesverbeteringen. Softwaremetrieken betreffende de kwaliteit van het softwareproces worden gebruikt voor kosten-batenanalyses op nieuw toegepaste technologie. Technologische (software)innovaties en methoden die resulteren in kwaliteitsverbeteringen worden als zodanig geïdentificeerd en gebruikt om veranderingen in het proces aan te brengen.

Defecteninformatie wordt gebruikt om defecten te analyseren en de oorzaken hiervan te onderzoeken. De bevindingen uit de oorzakenanalyse, in de vorm van leerprocessen, worden teruggekoppeld naar het desbetreffende softwareproces. Het doel hiervan is om het optreden van soortgelijke defecten in de toekomst tot een minimum te reduceren.

Samenvattend kan worden gesteld dat naarmate een softwareorganisatie *volwassener* wordt (een hoger gelegen CMM-niveau bereikt), softwareproducten tegen lagere kosten, met minder risico's en tegen hogere kwaliteit kunnen worden ontwikkeld.

SOFTWAREKWALITEITSKRINGLOOP

Een softwareproduct doorloopt verschillende fasen tijdens zijn levenscyclus. Afhankelijk van de fase waarin een softwareproduct verkeert en waarvoor men verantwoordelijk is, varieert de betekenis van productkwaliteit ([Trie94]). Nadat het softwareproduct op de markt is gebracht, komt het in de onderhoudsfase van zijn levenscyclus terecht. Tijdens de onderhoudsfase ondergaat het softwareproduct veranderingen, waarbij de kwaliteitseigenschap *onderhoudbaarheid* een grote betekenis heeft voor degene die onderhoud pleegt op het softwareproduct in kwestie.

De kwaliteitskringloop is een theoretisch model van de onderlinge wisselwerking tussen de kwaliteit van het softwareproces en de kwaliteit van een softwareproduct die in de verschillende fasen kan worden onderkend. In het bijzonder zal in deze paragraaf de onderhoudsfase nader worden belicht.

Kwaliteit, verschillende invalshoeken

'De kwaliteit van een object (bijvoorbeeld van een softwareproduct of IT-dienst) is het geheel van kenmerken van dat object die van invloed zijn op zijn vermogen, te voldoen aan vastgelegde en vanzelfsprekende behoeften' ([ISO94]). De invalshoek die wordt ingenomen bij het definiëren van het begrip kwaliteit bepaalt de manier waarop kwaliteit door de belanghebbende wordt beoordeeld. Garvin ([Garv84]) concludeert dat kwaliteit een complex en multidimensionaal concept is, dat kan worden beschreven vanuit vijf verschillende invalshoeken: (1) de transcendent, (2) de gebruikersgeoriënteerde, (3) de productiegeoriënteerde, (4) de productgerichte en (5) de waardegerichte invalshoek. Hieronder zullen alleen de gebruikers-, de productiegeoriënteerde en de productgerichte invalshoeken worden beschreven.

Gebruikersgeoriënteerde invalshoek

Juran ([Jura70]) definieert de gebruikersgeoriënteerde invalshoek van kwaliteit als 'fitness for use'. De klant verwacht dat het softwareproduct geschikt is voor gebruik. Deze geschiktheid wordt bepaald door de mate waarin het softwareproduct zijn primaire en/of secundaire bedrijfsprocessen ondersteunt. Ofwel de mate waarin de kwaliteitseisen, die vanuit de bedrijfsprocessen aan het softwareproduct worden gesteld, terug te vinden zijn in de productkarakteristieken van het product.

Nadat het softwareproduct bij de klant in gebruik is genomen, is het onderhevig aan allerlei krachten als gevolg van onder andere de continu veranderende bedrijfsprocessen. Het gemak en de kosten waarmee het softwareproduct aan de hierdoor veranderende kwaliteitseisen en bedrijfsprocessen aangepast kan worden, bepalen de blijvende 'geschiktheid voor gebruik' van de klant. Ofwel, de klant verwacht:

- *blijvende* betrouwbare werking van het softwareproduct;
- *blijvende* effectieve ondersteuning van het softwareproduct.

Productiegeoriënteerde invalshoek

Crosby ([Cros79]) definieert de productiegeoriënteerde definitie van kwaliteit als 'conformance to requirements'. Om een softwareproduct geschikt te maken voor gebruik moet aan de verwachtingen en behoeften van de klant worden voldaan. Deze kunnen in service level agreements (SLA) neergelegd zijn. De productiegeoriënteerde benadering stelt dat de kwaliteit van het ontwikkelingsproces wordt bepaald door de mate waarin aan deze verwachtingen wordt voldaan. Ofwel, de mate waarin een kwalitatief hoogwaardig eindproduct wordt opgeleverd in termen van bijvoorbeeld het aantal *latente defecten* en de hieraan verbonden kosten gedurende de onderhoudsfase. De productiegeoriënteerde benadering is terug te vinden bij het CMM; het (continu) verbeteren van het softwareproces heeft tot gevolg dat de kwaliteit van het software-eindproduct (tevens continu) verbetert ([Hump89]), ([Paul93]).

Productgerichte invalshoek

In tegenstelling tot de gebruikers- en de productiegeoriënteerde invalshoek die de kwaliteit van buitenaf beschouwen, wordt de productgerichte definitie van kwaliteit van binnenuit beschouwd. De productgerichte definitie van kwaliteit heeft betrekking op de intrinsieke productkarakteristieken die (in principe) kunnen worden gemeten ([Boeh84], [Gilb88]).

Vanuit de invalshoek die ingenomen wordt bij het definiëren van productkwaliteit kunnen verschillende verzamelingen kwaliteitseigenschappen door de belanghebbenden als relevant worden geacht. In de literatuur hebben diverse auteurs overzichten gegeven van wat zij wezenlijke eigenschappen van een softwareproduct vinden. Hierbij hadden de auteurs de intentie om een volledig en verifieerbaar beeld van de kwaliteit van een softwareproduct, voor alle belanghebbenden, te geven.

De kwaliteitsmodellen worden gerepresenteerd door een hiërarchisch stelsel van kwaliteitsattributen, waarbij elk lager niveau een verdere concretisering van het hogere niveau voorstelt. Deze modellen kunnen bijvoorbeeld worden gevonden bij [McCa77], [Boeh78], [Dele90] en [ISO92].

Kwaliteit in beweging

Het softwareproduct is op te vatten als het belangrijkste product van de softwareorganisatie. De productiegeoriënteerde benadering van kwaliteit stelt dat de kwaliteit van het ontwikkelingsproces bepaalt in welke mate een kwalitatief hoogwaardig eindproduct wordt opgeleverd ([Hump89], [Paul93]). Het kwaliteitssysteem van de softwareorganisatie, waarbinnen de kwaliteit van het softwareontwikkelings-

proces wordt bepaald, is een afbeelding van het kwaliteitsbeleid.

Bij het formuleren van het kwaliteitsbeleid streeft de softwareorganisatie naar een maximalisatie van bedoelde voorziene effecten: het op de markt brengen van een kwalitatief hoogwaardig softwareproduct. Nadat het softwareproduct op de markt is gebracht en bij de klanten in gebruik is genomen, komt de kwaliteit van het softwareproduct in beweging:

- als gevolg van storingen, veroorzaakt door defecten in het softwareproduct, ondervindt de klant problemen tijdens het gebruik van het softwareproduct;
- als gevolg van continue veranderingen in de bedrijfsprocessen neemt de mate waarin het softwareproduct de bedrijfsprocessen ondersteunt geleidelijk aan af.

Het kwaliteitssysteem van de softwareorganisatie is een afbeelding van het kwaliteitsbeleid.

De softwareorganisatie reageert op deze dalende kwaliteit door middel van het uitvoeren van diverse vormen van onderhoud: correctief, adaptief of perfectief onderhoud. De kwaliteit van de onderhoudsprocessen bepaalt de mate waarin de 'oorspronkelijke productkwaliteit' wordt gehandhaafd.

Verminderende kwaliteit

Er kunnen zich in onder meer de gebruikersorganisatie allerlei situaties voordoen die wijzigingen in het softwareproduct noodzakelijk maken. De softwareorganisatie zal door het uitvoeren van onderhoud een blijvende betrouwbare werking en blijvende effectieve ondersteuning van het softwareproduct garanderen. Door het herhaald uitvoeren van dit onderhoud neemt als gevolg van de 'Laws of Software Evolution' de kwaliteit van zowel het softwareproduct als het softwareproces geleidelijk aan af.

Laws of Software Evolution

Lehman ([Leh80]) heeft een groot aantal (complexe) softwareproducten onderzocht en heeft vervolgens een vijftal 'Laws of Software Evolution' geformuleerd. Twee van die wetten hebben gevolgen voor de kwaliteit van een softwareproces dan wel -product:

- *Wet van de voortdurende verandering.* De organisatie waarin het softwareproduct operationeel is, is continu onderhevig aan zowel interne als externe veranderingen. Om een blijvende effectiviteit van het softwareproduct te kunnen garanderen zullen

tevens veranderingen in het softwareproduct dienen te worden aangebracht.

- *Wet van de toenemende complexiteit.* Door het continu veranderen van het softwareproduct zal de oorspronkelijke structuur in complexiteit toenemen. Denk hierbij aan de legacy systemen die nog steeds bij de grotere ondernemingen operationeel zijn.

Op grond van deze twee wetten neemt zowel de kwaliteit van het softwareproduct als de kwaliteit van het softwareproces af. Mits de softwareorganisatie geen pogingen onderneemt om continu verbeteringen in zowel haar softwareproducten als -processen aan te brengen, zal de tevredenheid van de klant in de tijd bezien ook gaan afnemen.

Wet van de softwareaccumulatie

Diverse studies hebben uitgewezen dat complexe softwareproducten meer (correctief) onderhoud vereisen gedurende de onderhoudsfase ([Grem84], [Boeh81], [Lien81]). Complexe softwareproducten bevatten hierdoor meer *latente defecten*, die tijdens het ontwikkelingsproces onopgemerkt zijn gebleven. Figuur 1 laat zien dat aangebrachte wijzigingen dan wel opgeloste problemen nieuwe problemen kunnen introduceren en wijzigingen noodzakelijk kunnen maken ([Looij95], [Valla88]). Ofwel, door het uitvoeren van onderhoud kunnen nieuwe defecten worden geïntroduceerd. Onderhoud is dus moeilijk vanwege de toenemende complexiteit en als gevolg hiervan is meer onderhoud vereist.

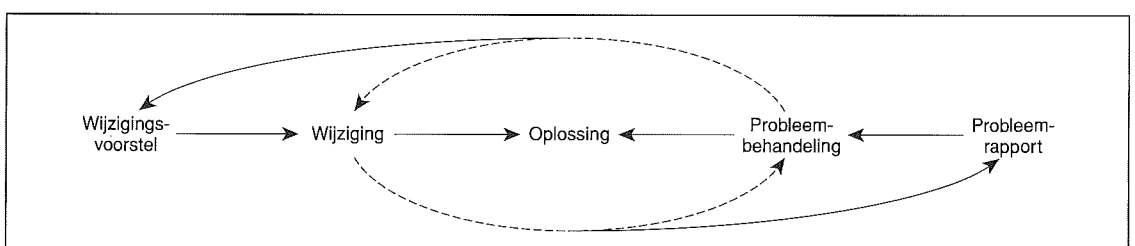
De reactie van de onderhoudsorganisatie op de dalende kwaliteit van het softwareproduct is op te vatten als een proces van *softwareaccumulatie*. Preventief onderhoud moet de neergaande spiraal van dalende kwaliteit doorbreken. Het doel van preventief onderhoud is tweeledig:

- het voorkomen van toekomstige problemen dan wel defecten;
- het verhogen van de kwaliteit van het softwareproduct.

(Continue) kwaliteitsverbetering

Om kwaliteitsverbeteringen te kunnen aanbrengen is het nodig het geleverde kwaliteitsniveau te registreren (in de vorm van onder meer probleem- en wijzigingsrapporten), defecten te analyseren en de oorzaken hiervan te onderzoeken. De bevindingen uit de oorzakenanalyse, onder andere in de vorm van leerprocessen, worden teruggekoppeld naar het desbetreffende kwaliteitssysteem of zelfs naar het kwaliteitsbeleid van de softwareorganisatie (zie figuur 2).

Kwaliteitsverbetering begint met het verkrijgen van inzicht in de wisselwerking tussen de kwaliteit van



Figuur 1. Wijzigings- en probleemschroef ([Looij95]).

het softwareproduct en de kwaliteit van het softwareproces ([Grad87], [Neil90]). Deze *enkelvoudige* kwaliteitskringloop is in deze paragraaf beschreven. Om de desbetreffende fase te kunnen bewaken en te beoordelen (dat wil zeggen het uitvoeren van kwaliteitscontroles op de software-entiteiten die in een softwareomgeving zichtbaar aanwezig zijn) en waar nodig verbeteringen door te kunnen voeren is een model nodig dat de bewuste fase in meetbare termen beschrijft ([Fent91]). Dit model zal in de volgende paragraaf worden beschreven.

KWANTITATIEF MODELLEREN VAN DE SOFTWAREKWALITEITSKRINGLOOP

Het softwareproduct doorloopt verschillende fasen tijdens zijn levenscyclus. Afhankelijk van de fase waarin het softwareproduct zich bevindt zal kwaliteit door de belanghebbenden uit verschillende aspecten zijn opgebouwd. Om deze aspecten te kunnen kwantificeren dient de softwareomgeving van de organisatie in meetbare termen te worden gemodelleerd ([Fent91]).

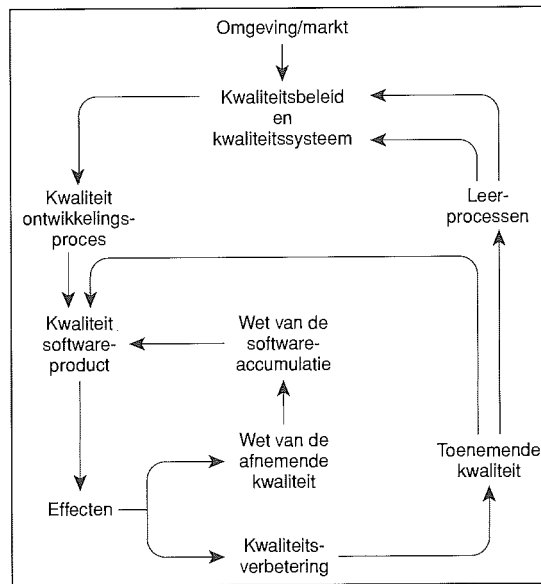
Softwareprocesmodel

Een organisatie dient haar toegevoegde waarde te maximaliseren door zowel kwalitatief hoogwaardige softwareproducten te leveren als haar softwareprocessen optimaal af te stemmen op haar omgeving. Om de klanten van een softwareproces optimaal te kunnen bedienen moet de organisatie de geleverde 'kwaliteit' van de in de softwareomgeving aanwezige software-entiteiten continu bewaken en beoordelen. Deze benadering is terug te vinden in het CMM; het (continu) verbeteren van de kwaliteit van het proces heeft tot gevolg dat de kwaliteit van het softwareproduct ook (continu) verbetert.

Fundamentele softwareprocessen

Alvorens het softwareprocesmodel zelf te behandelen dient een aantal reeds algemeen geaccepteerde aannames te worden gemaakt met betrekking tot dit proces ([Hump89]):

- Het softwareproces, dat opgedeeld kan worden in deelprocessen, kan worden gecontroleerd, gemeten en verbeterd.
- Door het gestructureerd uitvoeren van het softwareproces kunnen geplande kostenramingen, tijdsplanningen en kwaliteitsniveaus worden gehaald. Dit betekent dat de uitvoer van een softwareproces, binnen redelijke grenzen, kan worden voorspeld. Wanneer het softwareproces statistisch onder controle is, zullen herhaald uitgevoerde soortgelijke softwareprocessen dezelfde uitvoer opleveren.
- Wanneer de organisatie het softwareproces statistisch onder controle heeft, kunnen kwalitatief betere softwareproducten slechts worden verkregen door verbeteringen in het softwareproces aan te brengen. Als het softwareproces niet statistisch kan worden gecontroleerd, kunnen ook geen verbeteringen worden aangebracht.

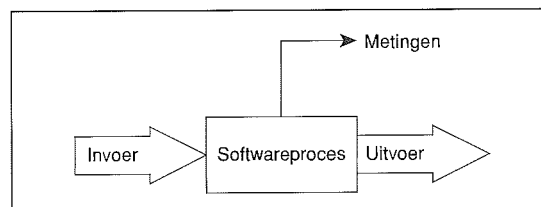


Figuur 2. Softwarekwaliteitskringloop.

Generiek softwareprocesmodel

Aangezien softwareprocesmodellen kunnen worden beschreven op een willekeurig abstractieniveau, dient gebruik te worden gemaakt van specifieke modelleermethoden en -technieken. Deze methoden en technieken dienen elementen, standaarden en structurele raamwerken aan te reiken waarmee door stapsgewijze verfijning gedetailleerde softwareprocesmodellen kunnen worden beschreven. De mate van gedetailleerdheid waarmee de softwareprocesmodellen kunnen worden beschreven, is afhankelijk van de *volwassenheid* van de organisatie die softwareproducten ontwikkelt en onderhoudt.

Figuur 3 toont een generiek softwareprocesmodel ([Hump89]). Het model laat zien dat het uitvoeren van een softwareproces, gegeven een bepaalde invoer en gebruikmakend van de hiervoor benodigde middelen, een bepaalde uitvoer oplevert. Het model uit figuur 3 kan zowel het gehele softwareproces voorstellen als slechts een gedeelte hiervan, bijvoorbeeld het testen van het softwareproduct. Door het aaneenschakelen van een aantal softwareprocesmodellen kan een logisch samenhangend softwareproces op willekeurig abstractieniveau worden beschreven.



Figuur 3. Generiek softwareprocesmodel.

Fenton ([Fent91]) stelt dat onderstaande drie categorieën software-entiteiten in een generiek softwareprocesmodel kunnen worden gemeten:

- *Softwareprocessen*. Tot een softwareproces behoren alle softwareactiviteiten die door de tijd heen plaatsvinden en die uitvoer creëren die van waarde is voor de klant (van het proces). Hierbij is een acti-

viteit een verzameling van gebeurtenissen die totstandkomt onder verantwoordelijkheid van één actor.

- *Softwareproducten.* Tot de softwareproducten behoren alle tastbare voorwerpen die voortgebracht worden door een softwareproces. Hiertoe worden dus tevens niet-softwareproducten gerekend. Voorbeelden van niet-softwareproducten zijn probleem- en wijzigingsrapporten, meetgegevens en handleidingen.
- *Middelen.* Tot de middelen behoren alle zaken die nodig zijn om een proces uit te kunnen voeren, met uitzondering van de (niet-)softwareproducten voortgebracht door een ander proces. Voorbeelden van middelen zijn personeel, softwaretools en computers.

Softwaremetrieken en softwareprocesmodel

Het gebruik van softwaremetrieken helpt een organisatie bij het eenduidig definiëren van de manier waarop een attribuut kan worden gemeten. Wel dient dan bekend te zijn van welke entiteit de metriek een indicatie geeft over de mate van bezit van een eigenschap. Een metriek dient gerelateerd te zijn aan een meetbare software-entiteit. Een en ander maakt het noodzakelijk om de softwarekwaliteitsmetrieken te integreren met het softwareproces ([Bhid90], [Busc90], [Fent91], [Henr96]).

In navolging van het door Fenton aangebrachte onderscheid in meetbare software-entiteiten kunnen softwaremetrieken in de volgende drie categorieën worden ingedeeld:

- *productmetrieken*, die attributen van het softwareproduct kwantificeren, bijvoorbeeld kosten, kwaliteit of grootte van een product;
- *procesmetrieken*, die attributen van het softwareproces kwantificeren, bijvoorbeeld kosten, doorlooptijd of aantallen voorkomen van een proces;
- *middelmetrieken*, die attributen van de middelen die nodig zijn om het proces te kunnen uitvoeren, kwantificeren, bijvoorbeeld kosten, kwaliteit of grootte van een middel.

Door de metrieken te relateren aan het softwareprocesmodel kunnen de volgende typen metrieken worden onderscheiden ([Bhid90]):

- *pre-procesmetrieken*, die attributen kwantificeren van software-entiteiten die als invoer van het softwareproces dienen;
- *in-procesmetrieken*, die attributen kwantificeren van software-entiteiten die tijdens de uitvoering van het softwareproces veranderingen ondergaan;
- *post-procesmetrieken*, die attributen van software-entiteiten kwantificeren betrekking hebbende op de uitvoer van het softwareproces.

nieert de softwareorganisatie hiertoe softwarekwaliteitsmetrieken die betrekking hebben op het geleverde kwaliteitsniveau van het softwareproces. Softwarekwaliteitsmetrieken die onderdeel uitmaken van softwaremetrieken kwantificeren kwaliteitsattributen van software-entiteiten.

Softwarekwaliteitsmetrieken kunnen, analoog aan bovengenoemde indeling, in drie categorieën worden onderverdeeld en in het model worden ondergebracht ([Kan95]). Door de relaties te onderzoeken tussen de pre-proces- en in-proceskwaliteitsmetrieken en de post-proceskwaliteitsmetrieken kunnen (kwaliteits)verbeteringen worden aangebracht in zowel de softwareprocessen als de softwareproducten. Hierbij dient rekening te worden gehouden met de karakteristieken van de softwareomgeving, en wel met name met het CMM-niveau waarop de organisatie zich bevindt.

Softwaremetrieken en volwassenheid van het proces

In de literatuur zijn vele software(kwaliteits)metrieken beschreven, zie bijvoorbeeld [Fent91], [Grad92] en [Kan95]. De hieruit gekozen verzameling metrieken bepaalt de mate waarin attributen van de binnen het softwareprocesmodel geïdentificeerde entiteiten kunnen worden gekwantificeerd.

Echter, attributen van de te meten entiteiten dienen wel 'zichtbaar' te zijn. De mate waarin attributen 'zichtbaar' zijn, wordt bepaald door de mate van *volwassenheid* van het softwareproces ([Pfle90]). Het CMM biedt hiervoor een goede context ([Hump89], [Paul93]).

Studies uitgevoerd door het Software Engineering Institute wijzen uit dat het merendeel van de softwareorganisaties zich op één van de eerste drie CMM-niveaus bevindt ([Sass96]). Hierom zullen alleen deze niveaus hieronder worden besproken.

CMM en softwaremetrieken

Het principe van de mate van *volwassenheid* van een softwareproces vindt zijn oorsprong in het CMM ([Hump89], [Paul93]). Naarmate een proces *volwassener* wordt, neemt de mate van gedetailleerdheid waarmee de organisatie haar softwareprocesmodel kan beschrijven steeds verder toe.

Softwaremetingen hangen nauw samen met *zichtbaarheid*; een organisatie kan slechts dat meten wat zichtbaar voor haar is. Het definiëren en analyseren van softwaremetrieken helpt een organisatie bij het begrijpen van die factoren die van invloed zijn op de *performance* en *capability* van het desbetreffende softwareproces. Het hierdoor verkregen inzicht maakt het voor een organisatie mogelijk het softwareproces steeds beter te kunnen definiëren, beheersen, controleren en (waar nodig) te sturen. Naarmate het proces *volwassener* wordt kunnen additionele metrieken worden gedefinieerd en geanalyseerd.

Derhalve dient de organisatie met het definiëren van metrieken te beginnen op niveau 1 om vervolgens stapsgewijs, door het incrementeel doorlopen van de *enkelvoudige* kwaliteitskringloop, additionele metrieken te definiëren zodat de organisatie uiteindelijk op niveau 5 uitkomt.

Een organisatie kan slechts dat meten wat zichtbaar voor haar is.

De softwareorganisatie dient kwaliteit tijdens de gehele levenscyclus van een softwareproduct in het oog te houden. Voor de verschillende fasen defi-

Initial software process

De uitgevoerde processen op dit niveau laten zich karakteriseren door een grote verscheidenheid aan bijvoorbeeld productiviteit, planningen en opgeleverde kwaliteit van het (intermediair) eindproduct. Een en ander is het gevolg van het ontbreken van structuur in en een matige definiëring van het softwareproces alsmede het ontbreken van controle- en stuurmaatregelen. Hierdoor zal het uitvoeren van softwaremetingen slechts ten dele (of helemaal niet) kunnen plaatsvinden ([Pfle90]). Organisaties die zich op dit niveau bevinden, dienen de nadruk te leggen op het aanbrengen van structuur in hun softwareprocessen en het definiëren van eisen waaraan de invoer en de uitvoer van een proces moeten voldoen.

Repeatable software process

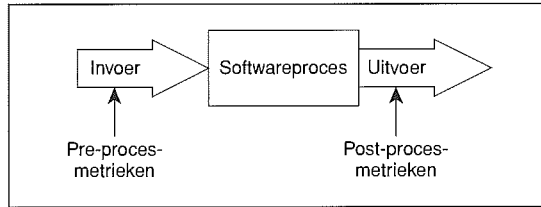
Op dit niveau zijn de invoer, de uitvoer en de middelen geïdentificeerd. De middelen die op dit niveau kunnen worden onderscheiden, zijn bijvoorbeeld personeel, softwaretools en computers. De organisatie heeft echter geen inzicht in de manier waarop de invoer wordt getransformeerd tot uitvoer. Dat wil zeggen, in de manier waarop een softwareproduct wordt ontwikkeld dan wel onderhouden. Figuur 4 toont het model van het *repeatable* softwareproces.

Figuur 4 laat tevens zien dat op dit niveau alleen metriecken kunnen worden gedefinieerd betrekking hebbende op de invoer, de uitvoer en de middelen, met uitzondering van softwareproductmetriecken ([Pfle90]). Fenton noemt de invalshoek die op CMM-niveau 2 wordt ingenomen om attributen van het proces te kunnen meten, de *externe invalshoek* ([Fent91]). Deze procesgeoriënteerde invalshoek richt zich op het meten van attributen die betrekking hebben op het softwareproces.

Defined software process

In tegenstelling tot niveau 2 bestaat op niveau 3 inzicht in het softwareproces. Op dit niveau kunnen de activiteiten waaruit het proces is opgebouwd, onderscheiden worden. Tevens zijn de eisen waaraan de onderlinge invoer en uitvoer van deze activiteiten moeten voldoen, bekend en gedefinieerd. Dit betekent dat de onderlinge invoer en uitvoer, die zijn op te vatten als intermediaire softwareproducten, beoordeeld en gemeten kunnen worden. Aan de op CMM-niveau 2 gedefinieerde verzameling metriecken kunnen, op dit niveau, productmetriecken worden toegevoegd ([Pfle90]).

Fenton noemt de invalshoek die op CMM-niveau 3 wordt ingenomen om attributen van het softwareproces te kunnen meten, de *interne invalshoek* ([Fent91]). Deze productgerichte invalshoek richt zich tevens op het meten van productattributen die van invloed zijn op de verschillende activiteiten (waaruit het proces is opgebouwd). Door het meten van productattributen kunnen voorspellingen wor-



Figuur 4. 'Repeatable' softwareproces.

den gedaan over hieraan gerelateerde procesattributen. Bijvoorbeeld, door het meten van de complexiteit van een softwareproduct kunnen voorspellingen worden gedaan over de inspanning die nodig is om een defect te kunnen lokaliseren ([Evan95]).

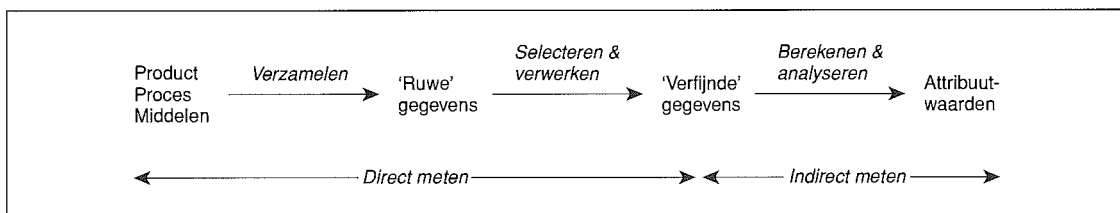
Softwaremetingen

Organisaties die, beginnende op CMM-niveau 1, een aantal malen de *enkelvoudige* softwarekwaliteitskringloop hebben doorlopen, komen uiteindelijk op CMM-niveau 2 terecht. Het bij CMM-niveau 2 behorende kwaliteitssysteem biedt voldoende basis om te beginnen met het definiëren van softwarekwaliteitsmetriecken die betrekking hebben op de *externe invalshoek* van het softwareproces.

Wanneer de organisatie door het uitvoeren van softwaremetingen iets wil weten over een bepaalde fase, dient te worden begonnen met het verzamelen van ruwe gegevens. Dit zijn gegevens die direct van een software-entiteit zijn af te leiden. Het verzamelen van deze ruwe gegevens is op te vatten als het laagste niveau van het direct meten van entiteiten. De directe softwaremetingen bestaan dan uit het maken van selecties op deze ruwe gegevens om verfijnde gegevens te kunnen verkrijgen.

Figuur 5 laat de rol van het verzamelen van *ruwe* gegevens zien in relatie tot het uitvoeren van softwaremetingen ([Mell92]). Het resultaat van een softwaremeting, verkregen door het toepassen van een geschikte softwaremetriek, is een aan een attribuut toegekende waarde. De verkregen waarde is afgeleid van de verfijnde gegevens, die op hun beurt zijn afgeleid van de verzamelde ruwe gegevens. Derhalve zijn alle softwaremetingen die een organisatie wil uitvoeren, afhankelijk van de ruwe gegevens die tijdens het uitvoeren van softwareprocessen kunnen worden verkregen en vastgelegd. In dit artikel zal niet worden ingegaan op waarborging van de kwaliteit van de gegevensverzekering en gegevensvastlegging. Voorbeeldmethoden hiervoor zijn te vinden bij [Grad87], [Pfle95] en [Kitc96].

Tevens kan uit figuur 5 worden afgeleid dat de hoeveelheid ruwe gegevens die tijdens het uitvoeren van softwareprocessen worden vastgelegd, afhankelijk is van de mate van detail en de reikwijdte waarmee de organisatie iets wil weten over het gele-



Figuur 5. Verband tussen indirecte en directe softwaremetingen ([Mell92]).

verde kwaliteitsniveau. Behalve door het CMM-niveau waarop de organisatie zich bevindt, wordt dit ook bepaald door de invalshoeken die door de belanghebbenden worden gekozen om 'kwaliteit' te kunnen beoordelen.

De binnen het CMM gedefinieerde metrieken zijn op te vatten als een hiërarchisch stelsel en zijn gerelateerd aan de *zichtbaarheid* en *volwassenheid* van het softwareproces. Door meerdere hiërarchisch geordende niveaus van metrieken te definiëren, kan elk hoger gelegen CMM-niveau worden gebruikt om steeds meer inzicht in en controle over het softwareproces en -product te krijgen ([Pfle90]). Naarmate het softwareproces *volwassener* wordt, dienen nieuwe softwaremetrieken te worden toegevoegd. Een dergelijke toevoeging is eveneens nodig als gevolg van nieuw verkregen inzichten dan wel behoeften. Derhalve helpen software(kwaliteits)metrieken een organisatie bij het incrementeel doorlopen van de softwarekwaliteitskringloop om uiteindelijk op CMM-niveau 5 te kunnen uitkomen.

METHODE VAN KWANTIFICEREN VAN DE SOFTWAREKWALITEITSKRINGLOOP

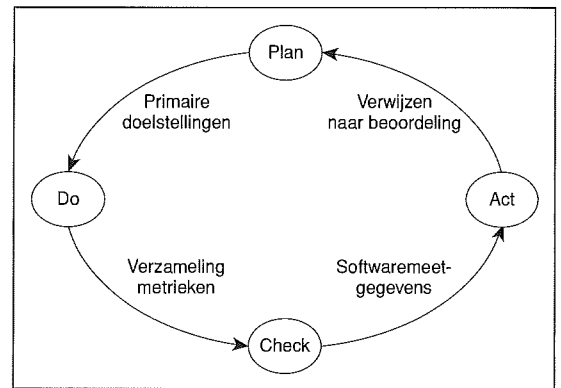
Om de kwaliteit van een softwareproces of -product te kunnen kwantificeren, dient de softwareorganisatie een geschikte verzameling metrieken te definiëren. Deze verzameling metrieken dient in overeenstemming te zijn met specifieke behoeften van de organisatie en karakteristieken van de softwareomgeving. De met het uitvoeren van softwaremetingen verkregen meetresultaten worden gebruikt om 'zichtbare kwaliteit' in de desbetreffende omgeving te kunnen bewaken en beoordelen. Tevens kunnen door het analyseren van de meetresultaten en de hiermee verkregen inzichten verbeteringen worden aangebracht in de desbetreffende software-entiteiten.

De softwaremeetmethode is toegepast op het correctieve onderhoudsproces van een softwareleverancier. Dit proces is verantwoordelijk voor het herstellen van defecten die door de klanten tijdens het gebruik met het softwareproduct worden geconstateerd.

De van de klanten afkomstige problemen worden door de helpdesk in een database vastgelegd. Volgens beoordeling door het probleembeheer kan een probleem worden veroorzaakt door een defect in het softwareproduct. Dit type problemen wordt vervolgens toegewezen aan het onderhoudsproces van de onderhoudsorganisatie. Op basis van de probleemgegevens wordt het defect (eventueel) opgelost en worden aanvullende gegevens over de aangebrachte wijziging in de database vastgelegd.

Shewart quality cycle

Binnen de CMM-filosofie wordt de 'Shewart quality cycle' als uitgangspunt genomen om (continu) verbeteringen te kunnen aanbrengen in zowel het softwareproces als het -product ([Demi82], [Hump89]).



Figuur 6. Shewarts 'Plan-Do-Check-Act cycle' ([Demi82]).

Figuur 6 laat zien dat een organisatie vier opeenvolgende fasen dient te doorlopen om (kwaliteits)verbeteringen aan te kunnen brengen:

1. *Plan*. Beoordelen van de ontwikkelings- en/of onderhoudsomgeving om probleemgebieden te kunnen identificeren en hieruit meetbare primaire doelstellingen te kunnen afleiden. De doelstellingen kunnen met behulp van het 'Goal-Question-Metric' (GQM)-paradigma worden uiteengeerafeld in softwarekwaliteitsmetrieken ([Basi88]).
2. *Do*. Beschrijven van het stelsel van softwarekwaliteitsmetrieken.
3. *Check*. Verzamelen van ruwe gegevens van de in het softwareprocesmodel geïdentificeerde entiteiten. Op basis van deze ruwe gegevens kunnen door het toepassen van de metrieken waarden aan attributen van de binnen het model geïdentificeerde entiteiten worden toegekend.
4. *Act*. Analyseren van de meetresultaten ter verkrijging van inzicht in de desbetreffende software-entiteiten. Op basis van deze bevindingen kunnen verbeteringen worden aangebracht. Tevens kunnen door deze analyses nieuwe probleem- en aandachtsgebieden worden geïdentificeerd.

Het softwaremeetproces is derhalve een systematische methode voor het meten, beoordelen en het (eventueel) aanbrengen van wijzigingen in het softwareproces door gebruik te maken van objectief verkregen gegevens. Dit is dus het toepassen van (geschikte) softwarekwaliteitsmetrieken om kwantitatieve en objectieve beoordelingen te kunnen maken over het geleverde kwaliteitsniveau van het desbetreffende softwareproces.

Plan: beoordelen softwareomgeving

Het doel van deze stap is om door het beoordelen van de softwareomgeving te komen tot een geschikte verzameling softwarekwaliteitsmetrieken waarmee het kwaliteitsniveau kan worden gemeten. Door het uitvoeren van een CMM-beoordeling kunnen probleem- en aandachtsgebieden worden vastgesteld en kan worden bepaald op welk niveau de organisatie zich bevindt.

Kwantitatief modelleren softwareomgeving

Het beoordelen van de mate van volwassenheid van het softwareproces in kwestie is de eerste stap in het bepalen van de te verzamelen metrieken ([Pfle90]). De beoordeling bestaat uit het in meetbare termen modelleren van de softwareomgeving ([Pfle93]). Het resultaat hiervan is een beschrijving van de software-entiteiten waarover de organisatie iets wil weten; het softwareproces, de softwareproducten en de middelen die nodig zijn om het proces uit te kunnen voeren.

Formuleren primaire doelstellingen

Softwaremetingen dienen te worden voorafgegaan door het formuleren van primaire doelstellingen ([Basi88], [Fent94], [Kitc96]). Door het formuleren van een stelsel doelstellingen wordt tegemoetgekomen aan de specifieke behoeften van de organisatie en derhalve aan de nuttigheidseis die onder anderen Humphrey stelt aan metrieken ([Hump89]).

Basili geeft hiervoor een aantal richtlijnen in termen van de bedoeling van, of de invalshoek waarmee en de omgeving waarin softwaremetingen worden uitgevoerd ([Basi88]):

- *Bedoeling.* De doelstelling van softwaremetingen is het karakteriseren, beoordelen, voorspellen of verbeteren van kwaliteitsattributen van software-entiteiten. Gegeven de proceskarakteristieken voor CMM-niveau 2 en 3 zullen softwaremetingen op CMM-niveau 2 een 'beoordelend' karakter hebben en op CMM-niveau 3 een 'verbeterend' karakter.
- *Invalshoek.* De invalshoek die wordt ingenomen bij het uitvoeren van softwaremetingen die betrekking hebben op het geleverde kwaliteitsniveau. Op

CMM-niveau 2 zijn de gebruikers- en de externe procesgeoriënteerde invalshoeken te onderscheiden, terwijl op CMM-niveau 3 tevens de productgerichte invalshoek is te onderscheiden.

- *Omgeving.* Een beschrijving van de omgeving waarin de software-entiteiten zich bevinden en waarvan de organisatie iets wil weten door het uitvoeren van softwaremetingen.

Uit bovenstaande richtlijnen kan worden afgeleid dat de te formuleren doelstellingen in overeenstemming moeten zijn met de *volwassenheid* van de softwareorganisatie. Immers, een organisatie kan slechts dat meten wat 'zichtbaar' voor haar is en het is derhalve zinloos om doelstellingen te formuleren die niet gekwantificeerd kunnen worden.

GQM-paradigma

De softwaremeetmethode gebruikt het GQM-paradigma om de geformuleerde doelstellingen te kunnen relateren aan softwaremetrieken. Dit paradigma stelt dat door het top-down uiteenrafelen van doelstellingen in kwantificeerbare vragen hieraan softwarekwaliteitsmetrieken kunnen worden gerelateerd ([Basi88]).

De te definiëren kwaliteitsmetrieken moeten zijn gerelateerd aan de binnen het softwareprocesmodel geïdentificeerde meetbare software-entiteiten. De kwantificeerbare vragen dienen dus betrekking te hebben op de entiteiten die 'zichtbaar' in een bepaalde softwareomgeving aanwezig zijn. Op die manier kunnen per doelstelling een aantal vragen en de hieraan gerelateerde softwarekwaliteitsmetrieken worden geformuleerd.

Het proces is hier op CMM-niveau 2 gedefinieerd en gekwantificeerd. In de onderhoudsomgeving van de softwareleverancier zijn de entiteiten 'problemen', 'defecten' en 'wijzigingen' te onderkennen. Gegeven de karakteristieken van het onderhoudsproces wordt in tabel 1 een opsomming gegeven van de uitwerking van de GQM-methode, waarbij rekening is gehouden met de volwassenheid van het onderhoudsproces.

Doelstellingen	Vragen	Metrieken
Beoordelen tevredenheid klant	Hoeveel problemen beïnvloeden de klant?	Incoming problem rate Fix quality Defect density Fix backlog
	Hoe lang duurt het voordat een probleem is opgelost? (vergeleken met verwachtingspatroon en SLA)	Fix response time % Delinquent fixes Fix Backlog
Beoordelen effectiviteit onderhoudsproces	In welke mate worden problemen opgelost?	Fix quality Fix backlog Defect density Incoming problem rate
Beoordelen efficiëntie onderhoudsproces	Waaraan worden de middelen besteed?	Maintenance staffing Problem type distribution Defect type distribution
	Hoe onderhoudbaar is het softwareproduct?	Defect density Fix quality Backlog management index

Tabel 1. Uitwerking GQM-methode voor het correctieve onderhoudsproces.

Do: beschrijven softwarekwaliteitsmetrieken

Het doel van deze stap is om de in het softwareprocesmodel geïdentificeerde metrieken te beschrijven. Deze beschrijving bevat onder andere, per metriek, de volgende elementen:

- *Waarom.* Dit gedeelte beschrijft waarom dient te worden gemeten. Het waarom vindt zijn neerslag in de doelstellingen die door de organisatie zijn geformuleerd. Ook kunnen de door de CMM-beoordeling geïdentificeerde aandachtsgebieden reden zijn om metingen te gaan uitvoeren.
- *Wat.* Naast een beschrijving van wat door de metriek wordt gemeten, wordt tevens een definitie van het te meten attribuut gegeven.
- *Waar en wanneer.* Door het beschrijven van een softwareprocesmodel kan worden aangegeven waar in het softwareproces metingen dienen te (kunnen) worden uitgevoerd. Tevens zullen sommige metingen een eenmalig karakter hebben, terwijl andere metingen een terugkerend karakter zullen hebben. Een en ander zal afhankelijk zijn van de doelstellingen, behoeften en karakteristieken van de softwareomgeving en zal in het meetprogramma expliciet moeten worden gemaakt.
- *Hoe.* Het hoe-gedeelte handelt over de manier waarop de metriek gemeten dient te worden, welke gegevens benodigd zijn om de metriek te kunnen berekenen en het tool waarmee de metingen uitgevoerd zullen worden.

- *Wie.* In het plan dient naar voren te komen welke typen analyses worden uitgevoerd, door wie de analyses zullen worden gedaan en hoe de analyses kunnen worden gebruikt in het besluitvormingsproces van de betrokken actoren.

De tijdens de planfase geïdentificeerde metrieken zijn in deze fase uitgewerkt. Het resultaat is een softwaremeetplan waarin, naast organisatiespecifieke aandachtsgebieden, de desbetreffende softwarekwaliteitsmetrieken zijn uitgewerkt.

Check: berekenen softwarekwaliteitsmetrieken

Om het geleverde kwaliteitsniveau te kunnen bewaken en te beoordelen dient de organisatie softwaremetingen uit te voeren. De organisatie maakt hierbij gebruik van de in de paragraaf 'Do' beschreven verzameling metrieken.

Act: beoordelen en terugkoppelen van de meetresultaten

Op basis van de verkregen meetresultaten kunnen aanbevelingen worden gedaan met betrekking tot het verbeteren van het softwareproces. In het kader van dit artikel voert het te ver om hier uitgebreid op in te gaan. Voorbeelden hiervan zijn terug te vinden bij [Grad92] en [Kitc96].

CONCLUSIE

Organisaties die softwareproducten ontwikkelen en/of onderhouden dienen kwaliteitssystemen te ontwerpen en in te richten die kwalitatief hoogwaardige softwareproducten kunnen voortbrengen. Uit de literatuur over softwareontwikkeling kan worden opgemaakt dat de jaren negentig zich laten kenmerken door de bewustwording in de softwareindustrie dat continue kwaliteitsverbetering van doorslaggevende betekenis is dan wel kan zijn om op de langere termijn te kunnen overleven.

Voor deze organisaties zijn kwaliteitssystemen beschreven, die hen ondersteunen in het continu kunnen doorvoeren van verbeteringen in hun softwareprocessen en -producten. Het Capability Maturity Model van Humphrey en Paulk is wellicht het bekendste.

Binnen het CMM wordt de 'Shewart quality cycle' als uitgangspunt genomen om (continu) kwaliteitsverbeteringen te kunnen aanbrengen in zowel het softwareproces als -product. Tijdens het incrementeel doorlopen van de kwaliteitskringloop kunnen softwaremetrieken de desbetreffende organisatie helpen om (continu) verbeteringen door te voeren; verbeteringen dienen gepaard te gaan met softwaremetingen. In de softwareliteratuur wordt veel aandacht besteed aan softwaremeetmethoden, softwaremetrieken en softwaremetingen.

De in dit artikel beschreven softwaremeetmethode maakt gebruik van Shewarts kwaliteitskringloop,

De in dit artikel beschreven softwaremeetmethode is toegepast op het correctieve onderhoudsproces van een softwareleverancier. Met behulp van deze methode werd de organisatie in staat gesteld om:

- het proces op meerdere niveaus te modelleren, waarbij elk lager gelegen niveau een verdere concretisering van een hoger gelegen niveau voorstelt;
- kan worden gespecificeerd waar, binnen het model, (niet)-softwareproducten worden gecreëerd dan wel worden gewijzigd;
- kan worden bepaald waar in het softwareprocesmodel metingen moeten worden uitgevoerd, door metrieken in het model te relateren aan attributen van meetbare software-entiteiten.

Met behulp van deze methode is een stelsel softwarekwaliteitsmetrieken gedefinieerd waarmee het door het onderhoudsproces geleverde kwaliteitsniveau kan worden gekwantificeerd. De meetresultaten boden het management inzicht in de door het onderhoudsproces geleverde kwaliteitsniveau; daarnaast zullen er stappen ondernomen worden om een softwareverbeteringsprogramma op te stellen (dat gepaard zal gaan met een softwaremeetprogramma).

In het verlengde hiervan wordt het management van de onderhoudsorganisatie in staat gesteld om met behulp van de verkregen meetresultaten:

1. de effectiviteit en efficiëntie van het correctieve onderhoudsproces kwantitatief te definiëren:
 - *effectief:* de mate waarin het correctieve onderhoud effect heeft: de mate waarin defecten in het softwareproduct worden opgelost;
 - *efficiënt:* de mate van effect of resultaat vanuit de door de onderhoudsorganisatie gebruikte middelen;
2. de onderhoudsverbetering c.q. -verslechtering dan wel achteruitgang in de onderhoudbaarheid van softwareproducten te identificeren en kwantificeren;
3. trends te identificeren om toekomstig onderhoud meer voorspelbaar te maken, zodat gekwantificeerde en betekenisvolle schattingen kunnen worden gemaakt. De met behulp van metrieken over een langere periode verkregen informatie kan als basis dienen om het onderhoudsproces naar CMM-niveau 2 te tillen.

Basili's GQM-paradigma, Humphreys CMM en Fentons softwareprocesmodelleermethode.

In tegenstelling tot de in de softwareliteratuur beschreven meetmethoden gebruikt de in dit artikel beschreven benadering een softwareprocesmodel en laat de methode verschillende (kwaliteits)invalshoeken toe. Hiermee wordt tegemoetgekomen aan de nuttigheidseis van de verzameling metrieken omdat ze in een praktische behoefte kunnen voorzien. Dit is van belang omdat vele software-attributen kunnen worden gemeten, terwijl de verkregen meetresultaten in sommige gevallen voor niemand van belang zijn. Tevens wordt de organisatie gedwongen om door het modelleren van het softwareproces na te denken over de componenten waaruit het proces is opgebouwd, zodat kan worden bepaald waar (binnen het model) softwaremetrieken kunnen worden gedefinieerd.

Samenvattend komt de beschreven softwaremeetmethode tegemoet aan de in de softwareliteratuur beschreven bevindingen:

- softwaremetingen moeten zijn gerelateerd aan de karakteristieken van de softwareomgeving en de behoeften van actoren binnen deze omgeving;
- de initiële verzameling kwaliteitsmetrieken waarmee softwaremetingen kunnen worden uitgevoerd, dient in beginsel 'klein' te worden opgezet; naarmate het softwareproces *volwassener* wordt, kunnen additionele metrieken aan deze verzameling worden toegevoegd;
- softwaremetingen dienen te zijn gerelateerd aan de *zichtbaarheid* en *volwassenheid* van een softwareproces;
- softwaremetingen maken een noodzakelijk onderdeel uit van kwaliteitsverbeteringen.

LITERATUUR

- [Basi88] V.R. Basili and H.D. Rombach, *IEEE Transactions on Software Engineering: The TAME project: Towards improvement-oriented software environments*, IEEE Software, June 1988, pp. 758-773.
- [Bhid90] S. Bhide, *Generalized software process-integrated metrics framework*, Journal of Systems and Software, December 1990, pp. 249-254.
- [Boeh78] B.W. Boehm, J.R. Brown, H. Kaspar, M. Lipow, G.J. MacLeod and M.J. Merritt, *Characteristics of software quality*, TRW Series of Software Technology, Vol. 1, North Holland Publishing Company, Amsterdam/New York/Oxford 1978.
- [Boeh81] B.W. Boehm, *Software engineering economics*, Prentice-Hall, Englewood Cliffs, New Jersey 1981.
- [Boeh84] B.W. Boehm, *Verifying and validating software requirements and design specifications*, IEEE Software, January 1984, pp. 75-88.
- [Bush90] M.E. Bush & N.E. Fenton, *Software measurement: A conceptual framework*, Journal of Systems and Software, December 1990, pp. 223-231.
- [Cros79] P.B. Crosby, *Quality is Free: the art of making quality certain*, McGraw-Hill, New York 1979.
- [Dele90] G.P.A.J. Delen en D.B.B. Rijsenbrij, *Kwaliteitsattributen van automatiseringsprojecten en informatiesystemen*, Informatie, januari 1990, pp. 46-55.
- [Demi82] W.E. Deming, *Out of the Crisis*, MIT Center for Advanced Engineering Study, Cambridge (MA) 1982.
- [Evan95] M.M. Evancho, *Modelling the effort to correct faults*, Journal of Systems and Software, April 1995, pp. 239-249.
- [Fent91] N.E. Fenton, *Software metrics: a rigorous approach*, Chapman & Hall, London 1991.
- [Fent94] N.E. Fenton, *Software measurement: A necessary scientific basis*, IEEE Transactions on Software Engineering, March 1994, pp. 199-206.
- [Garv84] D. Garvin, *What does product quality really mean?*, Sloan Management Review, Fall 1984, pp. 25-43.
- [Gilb88] T. Gilb and S. Finzi, *Principles of software engineering management*, Addison-Wesley, Reading (Mass.), 1988.
- [Grad87] R.B. Grady and D.L. Caswell, *Software metrics: Establishing a company-wide program*, Prentice Hall, Englewood Cliffs, New Jersey 1987.
- [Grad92] R.B. Grady, *Practical software metrics for project management and process improvement*, Prentice-Hall, Englewood Cliffs, New Jersey 1992.
- [Grem84] L.L. Gremillion, *Determinants of program repair maintenance requirements*, Communications of the ACM, August 1984, pp. 826-832.
- [Henr96] J. Henry, R. Blasewitz and D. Kettinger, *Research and Practice: Defining and implementing a measurement-based software maintenance process*, Journal of Software Maintenance, February 1990, pp. 79-100.
- [Hump89] W.S. Humphrey, *Managing the Software Process*, SEI Series in Software Engineering, Addison-Wesley Publishing Company, Reading (Mass.) 1989.
- [ISO92] ISO 9126, *Information Technology: software product evaluation: quality characteristics and guidelines for their use*, International Organisation for Standardization, Geneva 1992.
- [ISO94] ISO 8402, *Quality management and quality assurance: vocabulary*, International Organisation for Standardization, Geneva, 2nd edition, 1994.

Drs. C.M. Piek

Is sinds 1997 werkzaam bij KPMG EDP Auditors. In het kader van zijn afstudeeropdracht heeft hij bij een softwareleverancier onderzoek gedaan naar het meetbaar maken van softwareprocessen. Daarvoor is hij betrokken geweest als informatieanalist bij diverse soorten softwareontwikkelprojecten.

[Jura70] J.M. Juran and F.M. Gryna, *Quality planning and analysis: From product to development through use*, McGraw-Hill, New York 1970.

[Kan95] S.H. Kan, *Metrics and models in software quality engineering*, Addison-Wesley Publishing Company, Menlo Park, California 1995.

[Kitc96] B. Kitchenham, *Software metrics: Measurement for software process improvement*, NCC Blackwell, Cambridge, Massachusetts, USA, 1996.

[Leh80] M.M. Lehman, *On understanding laws, evolution and conservation in the large program life-cycle*, Journal of Systems and Software, January 1980, pp. 213-221.

[Lien81] B.P. Lientz and B.E. Swanson, *Problems in application software maintenance*, Communications of the ACM, November 1981, pp. 763-769.

[Looij95] M. Looijen, *Beheer van informatiesystemen*, Kluwer Bedrijfswetenschappen, Deventer 1995.

[McCa77] J.A. McCall, P.K. Richards and G.F. Walters, *Factors in software quality*, Rome Air Development Center, November 1977.

[Mell92] P. Mellor, *Failures, faults and changes in dependability measurement*, Journal of Information and Software Technology, October 1992, pp. 640-652.

[Neil90] M. Neil, R.J. Cole and D. Slater, *Research and Practice: Measures for maintenance management: a case study*, Journal of Software Maintenance, February 1990, pp. 223-240.

[Paul93] M.C. Paulk, B. Curtis, M.B. Chrissis and C.V. Weber, *Capability Maturity Model for Software, Version 1.1*, IEEE Software, July 1993, pp. 18-27.

[Pfle90] S.L. Pfleeger and C. McGowan, *Software Metrics in the Process Maturity Framework*, Journal of Systems and Software, December 1990, pp. 255-261.

[Pfle93] S.L. Pfleeger, *Lessons learned in building a corporate metrics program*, IEEE Software, May 1993, pp. 67-74.

[Pfle95] S.L. Pfleeger, *Maturity, models and goals: How to build a metrics plan*, Journal of Systems and Software, February 1995, pp. 143-155.

[Roch94] J.M. Roche, *Software metrics and measurements principles*, Software Engineering Notes, January 1994, pp. 77-85.

[Sass96] H. Sassenburg, G. Matser and P. Kazil, *Software Process Improvement: Waarom en wanneer?*, Informatie, juli/augustus 1996, pp. 50-57.

[Trie94] J. Trienekens, *Tijd voor kwaliteit, werken aan betere informatiesystemen*, Thesis Publishers, Amsterdam 1994.

[Valla88] S.R. Vallabhaneni, *Auditing the maintenance of software*, Prentice-Hall, Englewood Cliffs, New Jersey 1988.

Selectie van CASE-hulpmiddelen: een casebeschrijving

Drs. ing. R.F. Koorn CISA en J.C.J.M. Vrakking

De keuze van een CASE-hulpmiddel heeft een verstrekkende invloed op de wijze waarop in de erop volgende jaren de systeemontwikkeling plaatsvindt.

Het selectie- en implementatieproces kent een aantal essentiële stappen en valkuilen die tijdig en op hun juiste waarde moeten worden geschat.

INLEIDING

Systeemontwikkeling is geëvolueerd van een traditionele watervalmethode met derdegeneratieprogrammeertalen (3GL, zoals Cobol en C) naar een diversiteit aan ontwikkelmethoden met vierdegeneratietalen en hulpmiddelen (4GL). Deze ontwikkeling heeft zich geleidelijk voltrokken, stapsgewijs zijn er meer tekenpakketten, codegeneratoren, databaseontwerp- en testhulpmiddelen ingezet. Hierdoor ontstaat in organisaties veelal een palet aan niet-gestandaardiseerde en niet-communicerende hulpmiddelen die het gehele systeemontwikkelingsproces moeten ondersteunen. Een centrale database met ontwikkel- en onderhoudsgegevens voor de ontwikkelde applicaties (repository of data dictionary) is op deze wijze moeilijk te realiseren.

Externe omstandigheden kunnen een aanleiding vormen om te kiezen voor een nieuwe ontwikkelomgeving, bijvoorbeeld als informatiesystemen door wetswijzigingen of bedrijfstakontwikkelingen sneller gereed moeten komen (verkorten 'time to market') of als de ontwikkelafdeling marktconform moet opereren. Marktconform opereren betekent veelal het op kosten en doorlooptijd kunnen concurreren met externe softwareleveranciers om uitbesteding te vermijden. Het Jaar 2000 en de Euro vormen daarentegen meestal geen reden voor het ingaan van een selectietraject.

Geconfronteerd met deze situatie zijn organisaties op zoek naar geïntegreerde ontwikkelomgevingen. Dergelijke omgevingen van geautomatiseerde hulpmiddelen staan bekend als CASE (Computer Aided Software Engineering) of IPSE (Integrated Project Support Environment).

Een organisatie die dit selectietraject heeft doorlopen, is de Stigas GUO Groep (SGG). In dit artikel wordt het selectietraject van een CASE-hulpmiddel door SGG behandeld. Allereerst wordt een classificatie van ontwikkelhulpmiddelen beschreven. Na de doelstellingen en aanpak van het keuzeprocess worden de gehanteerde criteria behandeld. In de volgorde van de resterende selectiestappen zijn de keuzeresultaten en eerste praktijkervaringen weergegeven. Ten slotte wordt het gehele proces geëvalueerd, waarbij de aandachtspunten voor de EDP-auditor expliciet aan bod komen.

SGG verleent diensten op het gebied van sociale zekerheid en ARBO-zorg aan bedrijven, sectoren en sociale partners. Enige voorbeelden hiervan zijn het uitvoeren van sociale verzekeringswetten en pensioenregelingen; de begeleiding, bemiddeling en reïntegratie van arbeidsongeschikte en werkloze werknemers, het verlenen van ARBO-zorg en het aanbieden van (aanvullende) verzekeringsproducten. Bij SGG werken ruim 1800 medewerkers, waarvan circa zestig automatiseringsmedewerkers. Per 1 juli 1998 is een fusie van SGG met Cadans voorzien.

SGG beschikt over een landelijk datacommunicatienetwerk met vijftien AS/400-computers. Er wordt momenteel gebruikgemaakt van SDW en IDDOS.

CLASSIFICATIE GEAUTOMATISEERDE ONTWIKKELHULPMIDDELEN

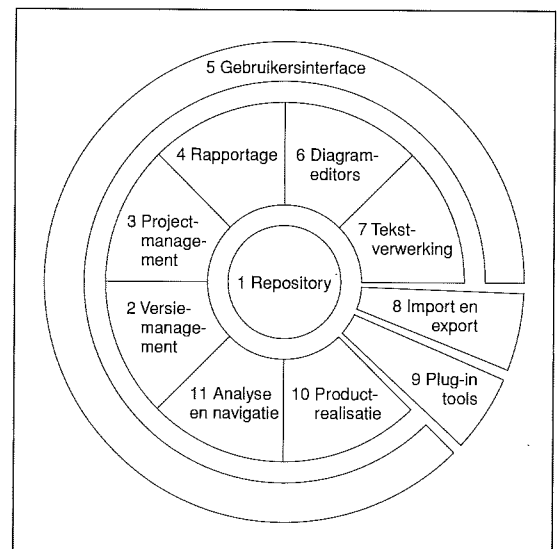
De op de markt aangeboden ontwikkelhulpmiddelen vertonen onderling grote verschillen. Toch worden bijna al deze producten onder de noemer 'CASE' aangeboden. Ze vertonen verschillen in de ondersteunde fasen van het ontwikkelproces, de methode(n) en technieken die ze ondersteunen en de wijze waarop dit gebeurt. Terwijl het ene hulpmiddel zich richt op informatieplanningsactiviteiten volgens de BSP-methode, richt een ander hulpmiddel zich op de procesgeoriënteerde analysemethoden als SADT of NIAM.

Ook de benodigde technische infrastructuur verschilt aanzienlijk: van een krachtige PC of werkstation tot aan minicomputers en mainframes.

De hulpmiddelen zijn op verschillende wijzen te classificeren; in tabel 1 is gekozen voor een indeling naar de mate van dekking van de ontwikkelfasen, doelgroepen en integratievorm¹. In de kolommen zijn de hulpmiddelen gerangschikt naar toenemende functionaliteit, van enkelvoudige 'tools' tot volledig geïntegreerde IPSE-omgevingen voor systeemontwikkeling en projectmanagement. Met de afwijkende kolomindeling is aangegeven dat de integratievorm niet één-op-één gelijk is met de indeling in de onderkende typen hulpmiddelen.

In de literatuur wordt naast bovengenoemde termen ook vaak een onderscheid gemaakt in 'front end' of 'upper-CASE' en 'back end' of 'lower-CASE'.

Met de term upper-CASE worden de verschillende hulpmiddelen bedoeld die het begin van het ontwikkelproces ondersteunen, dus hulpmiddelen voor informatieplanning, analyse en ontwerp. De term lower-CASE refereert aan de hulpmiddelen die ondersteuning bieden aan het realisatiedeel van het ontwikkeltraject. Hierbij kan onder meer worden gedacht aan codegeneratoren, testgegevensgeneratoren en faciliteiten voor knelpuntanalyse op performancegebied.



Figuur 1. Functionele aspecten van CASE.

Tabel 1. Classificatie van hulpmiddelen.

	Tool Individueel hulpmiddel	Toolkit Verzameling geïntegreerde hulpmiddelen	Workbench Eén ontwikkelomgeving	ICASE (Integrated CASE) Een geïntegreerde verzameling workbenches (en/of toolkits).	IPSE Geïntegreerde project- en ontwikkelomgeving
Levenscyclus In welke mate wordt het gehele ontwikkeltraject ondersteund?	Eén of meer afzonderlijke taken in het ontwikkelproces (binnen een fase)	Inhoudelijke activiteiten binnen één fase van het systeemontwikkelingstraject.	Meerdere fasen van de systeemontwikkeling	Alle fasen van de systeemontwikkeling	Volledige levenscyclus van een geautomatiseerd informatiesysteem (incl. gerelateerde activiteiten)
Doelgroep Wie zal de hulpmiddelen gebruiken?	Individuele ontwikkelaar		Meerdere projectleden	Hele projectteam (incl. projectmanager)	
Integratie In welke mate zijn de verschillende functies geïntegreerd?	Stand-alone. Hulpmiddel zonder koppelingen met andere hulpmiddelen	Geïntegreerde gegevens- en user interface. Communicatie via import- en export-faciliteiten en standaardbestandsformaten	Open architectuur. Hoge mate van integratie door middel van het gebruik van een centrale gegevensopslag binnen een workbench. Tussen workbenches onderling vindt communicatie plaats via import- en export-faciliteiten	Volledige integratie van hulpmiddelen. Integratie van inhoudelijke faciliteiten met functies voor het projectmanagement	Volledige integratie van hulpmiddelen. Hoge mate van integratie met het eindproduct (informatiesysteem incl. documentatie)

¹ Uit: Systeemontwikkeling in beweging, KPMG, 1991.

FUNCTIONALITEIT VAN CASE-PRODUCTEN

Het opdelen van CASE-producten in afzonderlijke functionele gebieden of componenten maakt het eenvoudiger invulling te geven aan dit begrip. In figuur 1² is een schematische weergave van CASE-producten

ten gegeven, onafhankelijk van de momenteel op de markt aangeboden hulpmiddelen. Zo kan op hoofdlijnen worden bepaald welke functionele gebieden bij de keuze van de ontwikkelomgeving van belang zijn. De in figuur 1 weergegeven hoofdfuncties van CASE-producten zijn in tabel 2 nader verklaard. Deze geeft een overzicht van de algemene functionaliteit van dergelijke producten.

Functie	Beschrijving functionaliteit
1. Repository	De repository bevat de metagegevens zoals zogenaamde 'business rules', die tijdens de diverse uitvoerende en projectbeheersende fasen van het ontwikkelproces worden gedefinieerd. Het is in feite het CASE-zenuwcentrum, waarin verschillende gebruikers terecht kunnen: informatieanalisten, databaseontwerpers en projectmanagers. Dit neemt het nadeel van afzonderlijke hulpmiddelen weg aangezien alle CASE-componenten via de repository kunnen communiceren. Voor de implementatie van de repository wordt steeds meer gebruikgemaakt van commercieel beschikbare relationele databasemanagementsystemen. Er is onderscheid te maken tussen een statische en een dynamische repository: een statische fungeert als registratiemiddel van ontwerpgegevens, een dynamische kan ook in de productieomgeving worden gebruikt door applicaties die hun gedrag kunnen aanpassen aan de 'business rules' in de repository.
2. Versiemanagement	Het moet in een CASE-hulpmiddel mogelijk zijn verschillende versies van dezelfde gegevens te bewaren en derhalve oude versies te hergebruiken. Het versiemanagement omvat het beheren van wijzigingen, het afhandelen van en rapporteren over problemen en het distribueren van productversies, in overeenstemming met de actuele inhoud van de repository. Versies kunnen zich in afgescheiden fasen c.q. omgevingen bevinden, zoals in de ontwikkel-, acceptatie- of productieomgeving.
3. Projectmanagement	Het plannen, begroten, sturen en bewaken van de voortgang van het ontwikkelproces vergt bij complexe projecten een goede ondersteuning. Hiervoor dient het hulpmiddel fasespecifieke gegevens te kunnen registreren over projectplanning en -bemensing, problemen, voortgang en dergelijke.
4. Rapportage	Vanuit de repository kunnen de ontwikkelaars snel de gewenste overzichten vervaardigen van ontwerpgegevens, zoals cross-references van programmamodules en gegevenstabellen. Naast de standaardoverzichten zijn projectplanningsoverzichten en ad hoc querymogelijkheden belangrijke faciliteiten.
5. Gebruikersinterface	Een consistente interface voor alle CASE-hulpmiddelen verkort de inleertijd en verhoogt de gebruikersvriendelijkheid. Een onderdeel van de interface is vaak de autorisatie tot de voor de ontwikkelaar relevante functies.
6. Diagram-editors	De gehanteerde – veelal vrij te kiezen – methoden en technieken functioneren met modellen en schemata. Voorbeelden zijn entiteit-relatiemodellen en dataflowdiagrammen. De diagrammen dienen op een snelle en consistente wijze te kunnen worden aangemaakt en worden onderhouden. De generatie kan deels automatisch geschieden op basis van bijvoorbeeld een ingevoerd datamodel.
7. Tekstverwerking	Deze faciliteit kan worden gebruikt voor het verwerken van pseudocode, het van commentaar voorzien van de programmatuur en voor het opbouwen van helpschermen. In de tekst kunnen relaties met gespecificeerde objecten worden gelegd.
8. Import- en export-faciliteiten	De uitwisseling met andere hulpmiddelen of met andere repository's waarborgt dat zonder hernieuwde invoer het ontwikkelproces kan worden vervolgd. Bij een projectteam kunnen de resultaten van de individuele ontwikkelaars op deze wijze worden gecombineerd. Bovendien levert een wisseling van CASE-omgeving geen desinvestering op. De standaardisatie bij deze vorm van uitwisseling is niet uniform; naast de diverse bestaande standaarden ³ tracht nu ook Microsoft haar repository als de-factostandaard te promoten.
9. Plug-in tools	Aanvullende functionaliteit is met modules (van derden) te realiseren. Hierbij is zowel de gegevensuitwisseling als de overeenkomstige gebruikersinterface essentieel. Een voorbeeld is een reverse engineering-hulpmiddel om uit de broncode een technisch model te genereren.
10. Productrealisatie	Routinematige ontwikkelhandelingen zijn met behulp van de repository te automatiseren met onder meer code- en databasegeneratoren, compilers, testfaciliteiten en performance-analyzers. Met behulp van prototyping zijn snel schermen en rapportages te creëren. De realisatie kan bijvoorbeeld na ontwikkeling op PC of AS/400 gebeuren voor meerdere doelplatformen: enerzijds een AS/400-applicatie in RPG en anderzijds een client/server-applicatie in C++ voor een PC- en Windows NT/Oracle-omgeving.
11. Analyse en navigatie	De consistentie van alle gegevenselementen, programmacode en schermen is te bewaken met speciale controlefuncties. Overtredingen van vastgelegde ontwerpregels zijn te traceren en te analyseren. De navigatiefunctie verplicht ontwikkelaars de te nemen stappen te doorlopen en de vereiste (mijlpaal)producten compleet op te leveren.

Tabel 2.
Hoofdfunctionaliteiten
CASE-producten.

² Uit: *Systeemontwikkeling in beweging*, KPMG, 1991.

³ Voorbeelden van standaarden zijn CDIF, ESE, PCTE, IRDS en ECMA.

HET PROJECT BIJ SGG

Alvorens het project te behandelen geven we eerst een korte beschrijving van de context waarbinnen het project wordt uitgevoerd.

Uitgangspositie van SGG

Op de afdeling Informatievoorziening van SGG werken 33 medewerkers bij Systeemontwikkeling en 28 bij Systeembeheer. SGG ontwikkelt met de beproefde SDM-2-methodiek.

Momenteel wordt het ontwikkelhulpmiddel SDW (System Development Workbench) gebruikt voor de fasen definitiestudie, basisontwerp en detailontwerp. Ook de functionele en technische gegevensmodellen worden in de SDW-repository opgenomen. De technische gegevensmodellen worden slechts ten dele geautomatiseerd vertaald in databaseattributen, de overige databasegerichte activiteiten gebeuren handmatig. Realisatie vindt plaats in Cobol en wordt ondersteund door de codegenerator IDDOS. Vervolgens vindt na de test- en acceptatiefasen overdracht naar de semi-productomgeving plaats. Hierin bevinden zich de geacordeerde programmaobjecten met de bijbehorende broncode. Vanuit deze omgeving vindt distributie van de programmaobjecten plaats naar de AS/400-computers in het land.

De sociale zekerheid is aan ingrijpende en elkaar snel opvolgende veranderingen onderhevig. Door de verdere afbouw c.q. herstructurering van de sociale zekerheid treden er momenteel bij SGG ingrijpende en frequente veranderingen op. Naast inhoudelijke wettelijke wijzigingen zijn ook de verdergaande privatisering en onderlinge samenwerking van uitvoeringsorganen bepalende factoren. Dit vereist onder meer het snel kunnen aanbieden van vervangende sociale verzekeringsproducten. Naast een kwalitatief goed product is derhalve de doorlooptijd van ontwikkel- en onderhoudsprojecten van eminent belang. De huidige op SDM gebaseerde ontwikkelomgeving is gezien de relatief traditionele en zware aanpak hiervoor niet altijd de meest optimale. Door externe ontwikkelingen staat het werken volgens de SDM-standaard regelmatig onder druk.

Opzet van de projectorganisatie

Het project 'Selectie CASE-tools' wordt uitgevoerd door een interne werkgroep in opdracht van de stuurgroep Projecten. In de stuurgroep Projecten zijn alle bedrijfsonderdelen van SGG vertegenwoordigd. In de werkgroep nemen deel:

- coördinator van de afdeling Informatievoorziening (projectleider);
- adviseur informatievoorziening;
- gegevensbeheerder;
- systeemontwerper;
- systeemanalist/programmeur;
- specialist technische infrastructuur.

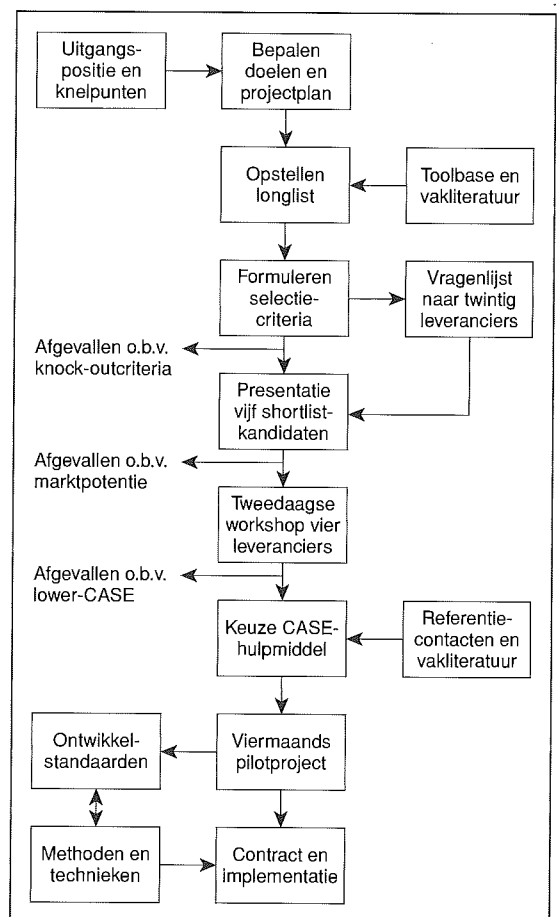
Met deze samenstelling is te verzekeren dat het keuzetraject vanuit de meest essentiële invalshoeken wordt beschouwd. Daar de ontwikkelaars ook module- en systeemtests uitvoeren is deze kennis ook

in de werkgroep vertegenwoordigd. Ontbrekend in de werkgroep is de invalshoek van de eindgebruiker van de met CASE ontwikkelde informatiesystemen. Hiervoor is bij SGG bewust gekozen, de CASE-selectie wordt voornamelijk gezien als een traject voor de afdeling Informatievoorziening.

HET SELECTIEPROCES

De hoofdstappen van het selectieproces zijn vergelijkbaar met die van de selectie van een standaardpakket. In figuur 2 zijn de fasen met de belangrijke elementen gevisualiseerd. Iedere stap is afgesloten met een mijlpaalproduct, waarover de stuurgroep een beslissing heeft genomen. Hierna volgen enkele kenmerkende zaken die bij dit proces een rol hebben gespeeld.

Naast een groot aantal overeenkomsten zijn ook verschillen tussen selectie van een CASE-product en van een standaardpakket te noemen. Een belangrijk verschil is dat de CASE-keuze een soort metakeuze is voor toekomstige informatiesystemen. Dit maakt het in de praktijk tot een automatiseringsintern project, waarbij de beslissingsonderwerpen veelal te specialistisch zijn voor het management. Zoals al blijkt uit figuur 2 kent de implementatie van een CASE-hulpmiddel de mogelijkheid een pilotproject uit te voeren en vervolgens voor een definitieve in-



Figuur 2. Selectiestappen.

voering te kiezen. Een dergelijke pilot met 'point of return' biedt meer opties bij matige resultaten dan een proefimplementatie met een ERP⁴-pakket. De belasting van een ERP-pakket voor medewerkers en processen maakt dat een organisatiebrede pilot van bijvoorbeeld vier maanden voorafgaande aan de finale beslissing over al dan niet implementeren vaak niet realistisch is.

BEPALEN VAN DE DOELSTELLINGEN

De belangrijkste doelen voor SGG om met een CASE-hulpmiddel te gaan werken zijn hieronder beschreven. Navolgende doelstellingen zijn kwalitatief geformuleerd; binnen SGG was men van mening dat de benodigde cijfers inzake de productiviteit of de verkorting van de doorlooptijd niet voldoende te onderbouwen zouden zijn.

Toename van de productiviteit

Uit ervaringen van andere organisaties en in de vakliteratuur is bekend dat ontwikkelhulpmiddelen bij een beheerst gebruik een productiviteitsverbetering opleveren. Voor een goede vergelijking van de productiviteit zonder en met een CASE-hulpmiddel is het beschikbaar over accurate en eenduidig te interpreteren ervaringsgegevens noodzakelijk. Er vond bij SGG een beperkte registratie plaats van ontwikkelgegevens, er is alleen sprake van functiepuntanalyse (FPA). De productiviteit uitgedrukt in bijvoorbeeld aantal functiepunten per uur was bij SGG voor de 3GL-situatie niet toereikend beschikbaar om te kunnen vergelijken met een situatie waarin vierdegeneratietalen worden toegepast.

Verkorten doorlooptijd projecten

Vanwege de eerder besproken dynamische markt-omgeving moet SGG op korte termijn aangepaste of geheel nieuwe verzekeringsproducten kunnen aanbieden. De gebruikte watervalmethode volgens SDM is minder geschikt voor deze snelle ontwikkelingen. In een tweetal projecten is daarom geëxperimenteerd met een RAD-achtige ontwikkelwijze (Rapid Application Development, gebaseerd op intensievere participatie van gebruikers en snellere terugkoppeling van tussenresultaten). De uitkomst van deze projecten was dat RAD als methode goed voldeed, maar dat het een toereikende geautomatiseerde ondersteuning ontbeerde. De resultaten waren niet voldoende snel aan gebruikers terug te koppelen. In de praktijk is gebleken dat zonder goede aansluiting tussen ontwikkelmethode en CASE-hulpmiddelen een efficiënte en beheersbare systeemontwikkeling nauwelijks of niet is te realiseren.

Geïntegreerde ontwikkelomgeving

Ondanks diverse standaardisatiepogingen is het aantal hulpmiddelen dat naadloos op elkaar aansluit nog beperkt. Het ontwikkelen met gebruikmaking van verschillende hulpmiddelen heeft als nadeel dat tussenproducten moeten worden overgedragen tussen niet op elkaar afgestemde hulpmiddelen. Tevens is zonder additionele interfaces of handmatige acties niet één ontwerpdatabase (repository) te creëren. Een repository heeft diverse voordelen, zoals:

- het waarborgen van de consistentie in het ontwikkelproces;
- het afstemmen van complexe gegevensstructuren van meerdere applicaties;
- het versnellen van het onderhoud;
- het genereren van programmaobjecten en actuele documentatie.

Interessante werkomgeving ontwikkelaars

In de huidige krappe en competitieve arbeidsmarkt is het noodzakelijk een uitdagende werkomgeving te creëren en te behouden. Het werken met een professioneel, grafisch georiënteerd CASE-hulpmiddel is voor veel dynamisch ingestelde softwareontwikkelaars aantrekkelijker dan het werken in een traditionele 3GL-omgeving.

Vanuit het perspectief van de ontwikkelaar bieden CASE-hulpmiddelen een goede ondersteuning van de werkzaamheden, waardoor repeterende werkzaamheden zoals het schrijven van standaardroutines en het uitvoeren van bepaalde consistentiecontroles grotendeels automatisch worden uitgevoerd. Tevens blijft hun 'employability' op peil, aangezien ze over nuttige kennis en vaardigheden beschikken voor andere werkgevers.

OPSTELLEN VAN DE SELECTIECRITERIA

De selectiecriteria zijn onderverdeeld in drie gradaties, namelijk:

- *knock-outcriteria*: aan deze criteria moest het hulpmiddel volgens SGG voldoen voor het realiseren van de doelstellingen en het verhelpen van de knelpunten van de huidige situatie;
- *eisen*: deze eisen zijn niet essentieel maar de benoemde functionaliteiten bieden belangrijke aanvullende mogelijkheden in het ontwikkelproces;
- *wensen*: alle overige door de werkgroep genoemde functionaliteiten. Op deze wijze werd in deze fase van het proces ieders inbreng gehonoreerd en zijn geen functies direct als irrelevant of te kostbaar terzijde geschoven.

In tabel 3 zijn enkele knock-outcriteria en belangrijke eisen vermeld die bij SGG zijn geformuleerd. Deze tabel geeft een indicatie van de criteria aan de hand waarvan een CASE-hulpmiddel is te beoordelen.

Een belangrijk selectie criterium is de verkrijgbaarheid van standaardbouwblokken bij een CASE-omgeving.

Om het selectietraject niet te compliceren is beperkt georiënteerd op toekomstige ontwikkelingen in de organisatie (fusie, workflowmanagementsysteem) of in de IT-sector (objectoriëntatie, Internet, Java, re-engineering). Deze ontwikkelingen zijn niet verstaald in functionele of technische eisen. Voor dit aspect is een grote waarde gehecht aan de visie en de R&D-inspanning van de leverancier.

⁴ Enterprise Resource Planning; bedrijfsbreed financieel, logistiek en productiesysteem.

Gebied	Zwaarte	Vraag	Reden
Algemeen	Knock-out	CASE-hulpmiddel genereert 'native' AS/400?	Presentatie, applicatieloga en gegevensafhandeling alle op AS/400 mogelijk (niet verplicht client/server-applicaties).
	Knock-out	Kent het versiemechanisme minimaal drie toestanden?	Scheiding van ontwikkel-, test- en productie-omgevingen.
	Knock-out	Client/server-ondersteuning volgens Gartner-model	Geschiktheid voor toekomstige systeemontwikkeling.
	Eis	Op welke niveaus is versiemechanisme toepasbaar?	Scheiding van individuele objecten, afgebakende gebieden en gehele repository.
	Eis	Welke zoekmogelijkheden kent het product?	De mogelijkheid om op diverse manieren te zoeken: via Windows, Hypertext, sleutelwoorden, contextgevoelige functie, etc.
	Eis	Welke autorisatieverfijning is aan te brengen?	Onderscheid naar individuele ontwikkelaars, teams en objecten/projecten.
Leverancier	Knock-out	Helpdesk in Nederland?	Terug kunnen vallen op expertise in Nederland in plaats van bij Amerikaanse fabrikant.
	Knock-out	Verhuur product tijdens pilot?	Ongebonden keuze na pilotevaluatie.
	Eis	Escrowovereenkomst af te sluiten?	Mogelijkheid tot eigen of leveranciersescrow.
Lower-CASE	Knock-out	Genereert standaardprogrammeertaal?	Voor het verminderen van de afhankelijkheid van de leverancier.
	Knock-out	Koppeling met bestaande AS/400-applicaties mogelijk?	Efficiënte gegevensuitwisseling is hierdoor mogelijk; geen noodzaak om bestaande applicaties in repository op te nemen.
	Eis	Reverse engineering van bestaande applicaties?	Hergebruik en onderhoud bestaande applicaties.
	Eis	Test- en debugfaciliteiten aanwezig of te koppelen?	Essentieel voor kwalitatief goede applicaties.
	Knock-out	Is een interface met SDW (of andere producten) beschikbaar?	Direct kunnen inlezen van functionele systeemmodellen in repository.
Upper-CASE	Knock-out	Welke ontwikkelmethoden worden ondersteund?	Mogelijkheid van waterval-, evolutionaire en objectgeoriënteerde methoden.
	Knock-out	RAD en prototyping ondersteund?	Prototyping vanuit technische én logische systeemmodellen biedt versnellingsopties.
	Eis	Ondersteuning AO-modellen?	Afstemming van organisatie, bevoegdheden en procedures op systeemfuncties en -autorisaties.

Tabel 3. Voorbeeld selectiecriteria.

Hanteren vragenlijst

SGG heeft als relatief onbekende organisatie haar jaarverslag met de uitgebreide vragenlijst (circa negentig vragen) meegestuurd om leveranciers te overtuigen van de omvang en financiële positie van een interessante klant. In selectietrajecten valt namelijk een vragenlijstmoeheid bij leveranciers te bespeuren; diverse potentiële klanten informeren bij een leverancier met een bijna uitputtende vragenlijst over de aanwezige CASE-functionaliteit en geboden ondersteuning. De vragenlijsten ontlopen elkaar eerder in structuur en diepgang dan in vereis-

te of gewenste functionaliteit. Het vergoed willen krijgen van de beantwoording is hierbij een opkomend fenomeen.

Uit de beantwoording van de vragenlijst is gebleken dat veel producten moesten afvallen omdat ze niet in staat waren 'native' Cobol- of RPG-programmatuur te genereren voor de AS/400. Deze producten zijn derhalve ten onrechte op de longlist gekomen.

Achteraf bleek dat vragen die niet geheel gesloten of concreet waren, vanwege de onbetrouwbare ant-

woorden geen nut in het verdere selectietraject hadden. Voorbeelden van nutteloze vragen waren:

– *Is prototyping mogelijk?*

Zelfs bij het meest traditionele ontwikkelhulpmiddel geeft de leverancier aan dat een vorm van prototyping toe te passen is. Welke vorm dit is en hoe dit precies gestalte moet krijgen, is uitsluitend in een demonstratie of workshop te achterhalen.

– *Zijn ontwikkeltemplates aanwezig?*

Deze zijn al dan niet aanwezig, maar onduidelijk blijft of ze specifiek voor jouw situatie toepasbaar zijn.

– *Zijn test/debugfaciliteiten aanwezig of te koppelen?*

Vrijwel alle leveranciers gaven hierop een positief antwoord, echter zonder eigen functies of producten te noemen maar door te wijzen op hun open repository.

– *Kan uw product interfacen met SDW?*

Hoewel ook hier het antwoord veelal bevestigend was, moesten de koppelingen door de meeste leveranciers nog worden gebouwd en worden getest.

– *Welke printfuncties kent uw product?*

De uitvoerige antwoorden gaven onvoldoende aan of de door SGG benodigde overzichten op het scherm en/of papier mogelijk waren.

– *Hoe groot is het R&D-percentages van de omzet?*

Dit percentage en het aantal betrokken R&D-medewerkers werd zonder uitzondering hoog opgegeven, zonder dat daaraan enige onderbouwing werd gegeven. Het is in feite een oncontroleerbare grootheid.

– *Hoeveel gebruikers gebruiken dit product?*

Tot lang in het selectieproces bleef onduidelijk of leveranciers hiermee het aantal ontwikkelaars, het aantal gebruikende organisaties (c.q. aantal licenties) of het aantal implementaties van verschillende versies dan wel op verschillende platformen bedoelden.

WORKSHOP EN KEUZE

Na de vergelijking van de CASE-producten met de knock-outcriteria resteerden vijf keuzekandidaten:

1. Pacbase van IBM;
2. KEY: Enterprise van Sterling software;
3. Adelia van de V&L Groep;
4. Obsydian van Synon;
5. Lansa van Aspect computing.

Pacbase en KEY: Enterprise boden hiervan als enige ondersteuning van het gehele ontwikkelproces. In eerste instantie is met deze twee producten een tweedaagse workshop gehouden. Vanwege de als onvoldoende getaxeerde marktpotentie of lower-CASE-functionaliteit vielen de bovengenoemde nummers 1 tot en met 3 af.

Aangezien geen unanieme keuze in de werkgroep kon worden bepaald, is besloten een pilot van twee dagen te organiseren om met de laatste twee producten, Lansa en Obsydian, een beperkte applicatie

te realiseren voor de AS/400. Daartoe is een technisch ontwerp gemaakt en aan de leveranciers is gevraagd deze applicatie gezamenlijk met SGG te realiseren. Deze applicatie bestond uit een beperkt aantal functionaliteiten, entiteiten en attributen. Extra onderdelen betroffen het vanuit de applicatie aanroepen van één van de bestaande AS/400-applicaties en het naderhand uitvoeren van een onderhoudsslag. Beide producten waren hiertoe in staat.

De daadwerkelijke keuze is vervolgens gebaseerd op een kortetermijn- versus langetermijnafweging. Op korte termijn sloot Lansa goed aan op de huidige manier van ontwikkelen. Het gebruik sluit nauwer aan bij de huidige werkwijze waardoor de inleertijd korter kan zijn dan bij Obsydian. Obsydian biedt als modern, op objectoriëntatie gebaseerd hulpmiddel meer doorgroeimogelijkheden. De initiële investering is echter hoger dan bij Lansa. Andere redenen voor de Obsydian-keus zijn geweest:

- de leverancier Synon is in Nederland en wereldwijd marktleider voor AS/400 CASE-hulpmiddelen;
- het product geschikt is voor meerdere doelplatformen en voor ontwikkeling van client/server-applicaties;
- het gebruiksgemak van een grafische omgeving;
- een aantal omvangrijke AS/400-gebruikers in Nederland gebruikt ook dit hulpmiddel (volgens opgave distributeur: Informatie Beheer Groep, TVM, CW Lease, NV Databank, Lease Plan, e.d.). Het aantal Nederlandse gebruikers is overigens voor de continuïteit en doorgroeimogelijkheden van beperkter belang dan het aantal gebruikers wereldwijd.

Met een aantal Nederlandse Obsydian-gebruikers is er contact over hun ervaringen in het selectie- en implementatieproces geweest, er zijn geen bezoeken afgelegd. Tevens is gesteund op het Gartner-onderzoek naar Obsydian⁵.

In de eindfase zijn naast de concrete functionaliteit ook subjectieve factoren een grote rol gaan spelen. Impliciet hebben enkele ontwikkelaars met een behoefte aan vertrouwde patronen gekozen voor de behoudende variant met Lansa. De oorspronkelijke weg van de selectiecriteria heeft in dit stadium niet de doorslag gegeven bij de keuze.

Het gevolg van de keus was dat Obsydian als lower-CASE-product met objectgeoriënteerde faciliteiten enigermate afweek van de oorspronkelijke lijst met eisen en wensen. Hierdoor ontbrak bijvoorbeeld ondersteuning van de definitiestudie.

De keuze voor een lower-CASE-hulpmiddel maakt ook dat SDW en Obsydian in samenhang zullen worden gebruikt. SDW zal niet langer het technisch ontwerp bevatten maar na het functioneel ontwerp entiteiten en attributen overdragen naar de Obsydian-repository. Het navolgend behandelde pilotproject moet uitwijzen hoe deze verschuiving van gebruikmaking van uitsluitend SDW naar deels SDW en steeds meer Obsydian zal plaatsvinden.

Contract

Voor de pilot is een partieel contract gesloten, waarbij de huurtarieven voor Obsydian en adviseurs zijn vastgelegd. Een definitief contract voor de aan-

⁵ Uit: Synon's Obsydian and 2E: Not just for the AS/400 anymore, Gartner, juni 1997.

schaf en het onderhoud wordt pas na een succesvolle afronding van de pilot opgesteld. Voordeel is dat de distributeur – met de leverancier op de achtergrond – in de pilot zijn beloften moet waarmaken, maar dit geeft als risico dat de distributeur vervolgens minder scherp hoeft te calculeren als hem bekend is dat hem de leverings- en implementatieopdracht zal worden gegund.

Het overgaan tot aanschaf na huurperiode voor de pilot biedt een go/no go-beslissing op basis van praktijkresultaten en houdt tevens de leverancier scherp.

Specifieke aandachtspunten die aan bod moeten komen in de contractonderhandeling voor een CASE-omgeving zijn:

- een exacte omschrijving van geboden ondersteuning in het inrichtingstraject (met name bij vaste prijsafspraken);
- de garantie dat de leverancier het hulpmiddel de komende jaren (stel minimaal vijf) blijft ondersteunen;
- de garantie dat de leverancier met het hulpmiddel alle courante OS/400-versies van IBM blijft ondersteunen;
- de reactie- én oplossingstermijn van fouten. De verhelping van fouten is bij voorkeur inclusief het herstel van hierdoor opgetreden inconsistenties in de repository;
- (Nederlandstalige) helpdeskondersteuning binnen bepaalde tijdslimieten;
- een escrowovereenkomst.

PILOT

Na formele accordering van de stuurgroep is een pilot met Obsydian gestart. Er is gekozen voor een project met een beperkt afbreukrisico, maar wel ten behoeve van het realiseren van een toepasbare applicatie. Deze pilot heeft als doel:

- opleiden van aantal medewerkers in Obsydian;
- vaststellen van een eerste versie van de standaarden voor SGG, te weten standaarden op het gebied van:
 - inrichting van objectomgeving van Obsydian;
 - naamgeving;
 - inrichting PC en AS/400;
 - gehele ontwikkeltraject;
- afwegen van kosten en baten;
- bepalen wat de gevolgen zijn voor de ontwikkelomgeving en technische infrastructuur;
- bepalen van de gevolgen voor medewerkers aan ontwikkel- en beheerszijde;
- uitdragen van de nieuwe werkwijze ('best practices' realiseren).

Pilotresultaten

De eerste karakteristieke resultaten van de pilot zijn kort aangegeven voor organisatorische, functionele en technische aspecten.

Organisatorisch

De manier van werken is innovatiever dan de huidige werkwijze; dit heeft een motiverende invloed op de pilotdeelnemers. Cruciaal is het dat het nieuwe CASE-hulpmiddel door de afdeling wordt gedragen. Middelen om dit te bereiken zijn het beleggen van een demoesessie voor de gehele afdeling, een wekelijkse statusbijeenkomst voor medewerkers en een periodieke informatiebrief. Daarnaast zijn er nog afzonderlijke bijeenkomsten met de projectleiders uit de gebruikersorganisatie.

Het beheer van de repository en van de standaardbouwblokken is van groot belang. Deze metagegevens en ontwikkelobjecten vereisen een centrale beheerfunctie, al dan niet geïntegreerd met het al ingerichte databasebeheer. Deze beheerfunctie is essentieel voor het optimale hergebruik van bouwblokken.

Op de automatiseringsafdeling bestaat nu het onderscheid tussen systeemontwerper en analist/programmeur. Dit onderscheid zal vervagen daar de analist/programmeur eerder bij het functionele deel van het project zal worden betrokken. Het ontwikkelproces zal door Obsydian veranderen. De verwachting is dat niet alle medewerkers dit veranderingsproces aankunnen. De gevolgen hiervan worden gecompenseerd doordat deze medewerkers bestaande applicaties kunnen blijven onderhouden. Er treedt bij SGG een vraageffect op waarbij projectleiders Obsydian te snel voor grote projecten willen inzetten. Dit houdt het gevaar in dat de CASE-invoering moeilijker beheersbaar wordt, omdat nog niet voldoende medewerkers zijn opgeleid, de standaarden nog niet zijn uitgekristalliseerd en dergelijke.

Met het CASE-hulpmiddel is het mogelijk zowel op een AS/400 als lokaal op een PC te ontwikkelen; periodiek zijn hiervoor down- en uploads van en naar de centrale repository nodig. Deze manier van werken brengt met zich mee dat stringent moet worden bewaakt dat niet meerdere ontwikkelaars dezelfde functies onderhanden hebben.

Het ontwikkelen van applicaties zou ook op afstand kunnen gebeuren. Binnen SGG zal worden overwogen of dit aan medewerkers wordt aangeboden. Gezien de huidige personeelsschaarste kan dit een aantrekkelijke flexibiliseringsoptie zijn.

Functioneel

Het voortraject van het ontwikkelproces, het nadenken over wat het product eigenlijk moet zijn en doen, is vanwege RAD van toenemend belang in het gehele ontwikkelproces. Zonder goed beeld en specificatie van het te realiseren product brengt een dergelijk CASE-hulpmiddel met gebruikmaking van RAD-technieken eerder het gevaar van kwalitatief ontoereikende en moeilijk onderhoudbare applicaties ('A fool with a tool is a bigger fool') met zich mee. Hierbij speelt de druk van gebruikers om een met RAD ontwikkeld prototype snel als volwaardige applicatie te kunnen gebruiken een rol.

Alvorens de realisatiefase start is het in een dergelijke CASE-omgeving essentieel dat op hoog niveau klassen met standaardobjecten zijn vastgesteld. Deze standaardklassen of bouwblokken worden gebruikt om er specificaties van te (over)erven.

Technisch

In de praktijk blijken eerder door de leverancier als eenvoudig afgeschilderde zaken toch complicaties te kunnen geven. Er zijn namelijk meerdere communicatievormen die goed dienen te functioneren. Het gaat daarbij ten eerste om de koppeling tussen Obsyidian en het netwerk. Hieruit blijkt dat de versies van met name CASE-hulpmiddel, OS/400, databasemanagementsysteem, PC- en netwerkbesturingssysteem op elkaar afgestemd dienen te blijven. Een ander belangrijk type koppeling, namelijk die tussen de nieuwe applicaties en bestaande Cobol-applicaties, is met het geteste CASE-hulpmiddel probleemloos aan te brengen. Dit vermijdt de noodzaak tot het opnemen van metagegevens van de bestaande applicatieportefeuille in de repository. Het eenmalig vastleggen van de bestaande structuren vergt namelijk een aanzienlijke inspanning.

Op performancegebied zijn twee invalshoeken van belang:

1. *Ontwikkelen en onderhouden van applicaties.* Door het kunnen overdragen van op PC's ontwikkelde applicaties is de belasting van de AS/400-computer sterk afgenomen. De op PC's aanwezige overcapaciteit wordt daardoor goed benut en er kan worden volstaan met lagere AS/400-investeringen.
2. *Laten functioneren van applicaties.* Vanwege de client/server-opzet kan het zwaartepunt van de applicatie verschuiven naar de PC-AS/400-communicatie en daarbij gebruikte middleware en databasemanagementsysteem. In veel organisaties worden hiermee performance- en stabiliteitsproblemen ondervonden.

Vervolg na pilot

Na een succesvolle pilot zal SGG overgaan tot een geleidelijke invoering in de diverse projectteams. De inrichting van Obsyidian moet dusdanig zijn dat duidelijkheid bestaat over de te gebruiken objecten en functies. In een systeemontwikkelingstraject zal pas van Obsyidian gebruik mogen worden gemaakt als bewezen is dat alle betrokken ontwikkelaars opgeleid zijn en bekend met de specifieke ontwikkelstandaarden en aandachtspunten. SGG zal bewaken dat ontwikkelaars door onbekendheid met een objectgeoriënteerde ontwikkelaanpak niet gaan pionieren. Er zal bijvoorbeeld ook een goede kennis van de standaard aanwezige bouwblokken moeten worden verkregen om te voorkomen dat al aanwezige functionaliteiten opnieuw worden ontwikkeld. Bij de werving van tijdelijke externe krachten zal voortaan mede worden geselecteerd op Obsyidianvaardigheden.

Indien de pilot tegenvallende resultaten oplevert is bij SGG geen noodscenario aanwezig. Uit praktijkervaring blijkt dat dan twee alternatieve oplossingsrichtingen voorkomen: aanpassing of opnieuw kiezen.

In het eerste geval poogt de organisatie met organisatorische en procedurele aanpassingen de ontwikkelaanpak dusdanig te veranderen dat selectief delen van het CASE-hulpmiddel kunnen worden ingezet. In het tweede geval wordt het CASE-hulpmiddel terzijde geschoven, waarna er een periode van handha-

ven en ad hoc verbeteren van de huidige werkwijze volgt. Deze bezinningsperiode kan enkele maanden tot enkele jaren duren tot het draagvlak weer aanwezig is om een grote CASE-stap voorwaarts te zetten. Slechts in incidentele gevallen implementeert de organisatie direct de tweede kandidaat uit de selectie.

Een CASE-omgeving creëert een nieuwe functie in de ontwikkelorganisatie: object- en repositorybeheerder.

De gevolgen voor ontwikkel- en gebruikersorganisatie zijn verstrekkend indien pas na diverse projecten blijkt dat het CASE-hulpmiddel niet voldoet. Reeds geïmplementeerde applicaties moeten veelal in een CASE-omgeving worden onderhouden waarvoor geen onderhoudscontract meer geldt. Dit is te ondervangen door alle relevante metagegevens naar de nieuwe repository te converteren. Lopende of net te starten projecten ondervinden een substantiële vertraging door het overschakelen van hulpmiddel. Al ontwikkelde delen zullen eventueel opnieuw moeten worden gebouwd. De ontwikkelaars en beheerders worden geconfronteerd met een nieuw opleidingsprogramma voor CASE, middleware en specifieke standaarden en procedures.

EVALUATIE VAN SELECTIEPROCES

Voor gelijksoortige selectie- en implementatietrajecten zijn attentiepunten te vermelden over de procesmatige en de inhoudelijke aspecten. De nadruk ligt hierbij op de selectie- en invoeringsfasen; de relatie tussen CASE-hulpmiddel enerzijds en ontwikkelproces en ontwikkelde applicaties anderzijds wordt beknopt behandeld. Een EDP-auditor zal onder andere onderstaande kritische factoren tijdens een evaluatie aan de orde moeten stellen.

Ter lering en ter vermaak zijn vier casussen opgenomen waar sprake was van valkuilen bij CASE-selectietrajecten.

Selectie

De doelen en scope van het selectietraject zijn veelal niet in meetbare eenheden uitgedrukt. De keuze en aanschaf is zonder geconcretiseerde doelen niet eenvoudig te rechtvaardigen op basis van de productiviteitstoename, versnelling van het ontwikkeltraject, lagere onderhoudskosten en dergelijke. Ook de terugverdientijd en afschrijvingstermijn zijn op deze wijze niet te bepalen. Sommige organisaties pogen met vergelijkingen tussen oude en nieuwe FPA-metrieken enige onderbouwing aan een kosten-batenanalyse te geven. De eerste projecten geven vaak een vertekend beeld door enerzijds een plannings- en budgetoverschrijding vanwege de gewennings-tijd en anderzijds een verhoogde productiviteit vanwege het feit dat de werkzaamheden tijdens het project intensief worden gevolgd⁶. Er blijkt in de praktijk een onderschatting van de organisatorische invoerings- en beheerkosten te zijn.

⁶ Dit wordt wel het Hawthorne-effect genoemd naar een onderzoek waarin de productiviteit van fabrieksarbeiders sterk steeg vanwege het feit dat er metingen werden verricht.

Haasje over

Bij een grote informatieverwerkende organisatie staat men voor een ingrijpend vernieuwingsproject van het primaire bedrijfssysteem. De keus van het al geselecteerde CASE-hulpmiddel wordt ter discussie gesteld. Het onderzoek naar dit hulpmiddel wijst uit dat het risicoprofiel hoog is en de opbrengsten pas na een kostbare implementatie in de onderhoudsfase zijn te realiseren. De geconstateerde risico's zijn onder andere:

- vele nieuwe CASE-functionaliteiten;
- beperkte aansluiting van product op doelomgeving;
- geen goede gedistribueerde gegevensverwerking;
- matige leveranciersondersteuning.

De onderkende risico's zijn individueel draagbaar, maar in combinatie niet. Besloten is een nieuw, kortdurend selectieproject uit te voeren opdat de belangrijkste delen van het nieuwe informatiesysteem in de nieuwe CASE-omgeving zijn te ontwikkelen.

Casus 1.

De gehanteerde knock-outcriteria, eisen en wensen waren geheel afkomstig van de ontwikkelafdeling; systeembeheerders en gebruikers zijn hierbij niet betrokken. Alhoewel het een ontwikkelgereedschap betreft zijn er wel degelijk indirecte eisen aan de CASE-omgeving te stellen vanuit:

- de gebruikers: prototyping, gebruikersinterface, AO/IC-ondersteuning, performance, documentatie;
- het rekencentrum: stabiliteit, databasebeheer, communicatie- en middlewarestandaarden, toekomstige platformen.

Deelname van vertegenwoordiging van deze beide groepen kan zorgen voor het maken van een gebalanceerde afweging. Een gebruikersvertegenwoordiger kan bijvoorbeeld aangeven de wachttijd tot een bruikbare toepassing te lang te vinden. Wordt een dergelijk geluid niet vroegtijdig gehoord, dan kunnen de ontwikkelaars te snel het CASE-product gaan inzetten, zonder dat al sprake is van een stabiele, gestandaardiseerde ontwikkelomgeving.

Direct met de drie marktleiders starten is een betere benadering dan het aanleggen van een extensieve longlist met alle verkrijgbare CASE-producten.

De belangrijkste leerervaring van het hier beschreven traject vormt de marktbrede benadering van leveranciers met een uitgebreide vragenlijst. Dit vergt een behoorlijke inspanning van de betrokken ontwikkelaars en verlengt de doorlooptijd. Het geeft daarentegen wel een goed beeld van de CASE-markt en technische ontwikkelingen. Uit een uitgebreide set eisen en wensen blijkt een hoog ambitieniveau, maar die set wijst ook op het bestaan van het gevaar van functionele 'overkill'.

Een snel en resultaatgericht alternatief is het overslaan van de longlistfase en direct de knock-outcriteria met de producten van bijvoorbeeld drie wereldwijde marktleiders vergelijken. Deze producten

zijn zo nodig aan te vullen met één of enkele nieuwe en veelbelovende producten, eveneens van marktleiders. Een – in de praktijk werkzaam gebleken – alternatief is het op de periodiek gehouden IT-beurzen met enkele ontwikkelaars gericht met de opgestelde eisenset de longlist terugbrengen tot een short list.

Op beide voorgaande wijzen is spoedig op een workshop aan te sturen als aan de knock-outcriteria wordt voldaan en de continuïteit en visie van de leverancier bevredigend overkomen. De ontwikkeling van het hulpmiddel dient in lijn te zijn met:

- *Gebruikersorganisatie.* Sluit de wijze van ontwikkelen aan op het type informatiesystemen, eisen, cultuur en geplande ontwikkelingen (fusie, procesgerichte organisatiestructuur, introductie workflow-management)?

- *Ontwikkelorganisatie.* Worden de door de organisatie gewenste toekomstige methoden en technieken ondersteund? Is het hulpmiddel geschikt voor alle applicatiebouw en -onderhoud, inclusief beslissings-ondersteunende systemen en end-user-computing of dient hiervoor flankerend beleid te worden gevoerd?

- *Verwerkingsorganisatie.* Is voorzien in faciliteiten ten behoeve van Internet, gedistribueerde verwerking en dergelijke? Zijn automatische back-up en recovery van de repository mogelijk? Hoe eenvoudig is het beheer van het CASE-hulpmiddel en van de repository? Kan met het CASE-hulpmiddel op én voor andere platformen worden ontwikkeld? Hiervoor dient dan bij voorkeur geen aanvullende programmatuur en middleware noodzakelijk te zijn, want daardoor ontstaan nieuwe afhankelijkheden. Zijn standaardkoppelingen op repository-niveau aanwezig om maatwerk te vermijden?

Goed product, slechte strategie

Het CASE-product werd bij een verzekeringsmaatschappij als kwalitatief goed beschouwd. Twee jaar na de start met het product besloot de leverancier niet verder in het product te investeren. Dit vanwege de beperkte marktpenetratie en het financiële perspectief. Te laat heeft de leverancier gekozen het product voor meerdere platformen geschikt te maken. De volgende keuze was voor het toekomstvaste product van een CASE-marktleider.

Casus 2.

Het richten op de marktleiders wordt ook door het onderzoeksbureau Gartner aanbevolen gezien de te verwachten 'shake-out' op de drukbezette markt van CASE-hulpmiddelen⁷. Tevens vindt er een convergentie van 4GL-, OO- en ICASE-producten plaats. Bovendien moet de CASE-omgeving een strategisch product voor de leverancier zijn om situaties zoals met van de markt gehaalde CASE-hulpmiddelen te vermijden. Dit is bijvoorbeeld gebeurd bij alle organisaties die gebruikmaakten van het Sybase-hulpmiddel Gain Momentum. Een verplichte overstap betekent een arbeidsintensieve overheveling van repositorygegevens. Hierdoor zien de meeste organisaties van een dergelijke overstap af en zal eventueel met meerdere, gekoppelde repository's worden vervolgd.

⁷ Uit: *Enterprise Applications Development Tools: Evaluation and Selection*, Gartner, september 1997.

Indien een organisatie tot de conclusie komt dat geen enkel hulpmiddel voldoet moeten ofwel de knock-outcriteria worden aangepast ofwel moet worden besloten vooralsnog af te zien van CASE-aanschaf. Indien geen van de marktleders voldoet aan de criteria kan dat aangeven dat de organisatie zich unieker voordoet dan zij in werkelijkheid is of dat er weinig verbetering te verwachten valt ten opzichte van het huidige gebruikte CASE-hulpmiddel.

Een aantal CASE-omgevingen kan worden aangevuld met referentiemodellen en standaardbouwblokken van derden-leveranciers. Algemeen toepasbare functionaliteit is snel organisatiespecifiek te maken. Hiervoor wordt de term 'componentware' gebruikt.

Het contract voor het uitverkozen CASE-product bevat in sommige situaties openeindregelingen of ongunstige bepalingen bij geschillen. Door uit te gaan van modelcontracten is dit bezwaar te onderwerpen. De prijsafspraken dienen bij voorkeur voor de pilot te worden overeengekomen. Na een succesvolle pilot verkeert de leverancier in een sterke onderhandelingspositie; de noodzaak om scherp te offeren is afgenomen gezien de voorkeur van de organisatie. Overigens wordt dit nadeel in Nederland gereduceerd aangezien meerdere distributeurs hetzelfde CASE-product aanbieden.

Driemaal = scheepsrecht

De transportorganisatie heeft in derde instantie succesvol een CASE-omgeving kunnen opzetten. Twee eerdere introducties mislukten. Bij de eerste invoering was de ontwikkelorganisatie nog niet gereed en ingericht om met dergelijke hulpmiddelen te starten. Programmeurs vielen vanwege onvoldoende kennis en ondersteuning terug in hun oude werkwijze. De tweede poging strandde vanwege een negatieve evaluatie van het allesbepalende, maar te hoog gegrepen pilotproject. De leverancier onderkende te laat de projectknelpunten. De derde invoering wordt structureel aangepakt. Hierbij blijkt dat het draagvlak intussen voldoende is voor een geslaagde implementatie.

Casus 3.

De waarheidsgetrouwheid van de beantwoording van de vragenlijst is te verhogen door aan te geven dat de beantwoording onderdeel uitmaakt van het uiteindelijke contract.

Invoering

Een invoering is gebaat bij een risicoanalyse vooraf. Met deze analyse is vast te stellen welke knelpunten zijn te verwachten op het gebied van:

- *Functionaliteit.* Zijn er inrichtingsbeperkingen aanwezig bij het gebruikmaken van de eigen methoden en technieken? Is bij de implementatie een fasering toe te passen? Komen er twee onderhoudsmethoden doordat de metagegevens van bestaande applicaties niet in de repository worden opgenomen?

- *Technisch.* Bestaat er kans op performance- of koppelingsproblemen? Hoeveel handmatige activiteiten zijn nodig bij het genereren van schermen, code en databasetabellen?

- *Organisatorisch.* Voorzien de opleidingen en begeleiding in de opleidingsbehoefte en beschikken de medewerkers over voldoende veranderingsbereidheid in verband met verschuivende taken?

Bij SGG moeten de ontwikkelstandaarden en werkwijze tijdens de pilot uitkristalliseren. Dit wordt ook wel 'method engineering' genoemd. De inrichting van de CASE-omgeving en van het ontwikkelproces wordt dus sterk bepaald door het pilotteam, type applicatie en het CASE-hulpmiddel. Deze situatie-specifieke invulling hoeft niet automatisch voor andersoortige, bijvoorbeeld omvangrijke, applicaties te voldoen.

Pas na enkele projecten van verschillende complexiteit zal enige stabilisering van ontwikkelaanpak en CASE-gebruik optreden.

De projectleider heeft een onmisbare rol in het uitdragen van de CASE-voordelen, ook voor de ontwikkelaars die niet betrokken zijn bij pilot of eerste paar projecten. In tegenstelling tot een implementatie van een ERP-pakket zal pas na enkele projecten van verschillende complexiteit enige stabilisering van ontwikkelaanpak en CASE-gebruik optreden. Er is derhalve meer sprake van een programmamanager dan van een projectleider.

SGG heeft bij de pilot een adviseur van de leverancier betrokken. Dit is ook de intentie voor het invoeringstraject. Deze adviseur heeft belang bij het welslagen van de pilot gezien de nog definitief te maken keuze; hij/zij zal zijn taak eerder zien als een resultaat- dan een inspanningsverplichting.

Ontwikkelproces

De ondersteuning van de systeemontwikkeling met een CASE-hulpmiddel kan een object van onder-

Bijna klaar

Een andere verzekeringsmaatschappij wordt tijdens haar selectietraject geconfronteerd met de nieuwste, sterk verbeterde versie van een CASE-product. Vanwege extra functionaliteiten wordt door de projectgroep aan dit product de voorkeur gegeven. Tijdens de pilot blijkt echter dat de stabiliteit van het product te wensen overlaat. Bovendien moet bij vrijwel ieder supportverzoek de leverancier in Amerika worden ingeschakeld, de Nederlandse helpdesk degradeert zich door kennisachterstand tot – vertragend – doorgeefluik. Na het overwinnen van diverse kinderziekten is de ontwikkelafdeling pas na anderhalf jaar tevreden over de keuze.

Casus 4.

Drs. ing. R.F. Koorn CISA
Is manager bij KPMG EDP
Auditors en is gespecialiseerd
in de advisering over en evaluatie van informatie-infrastructuren en systeemontwikkelprojecten. Hij werkt vanaf januari 1998 bij KPMG te San Francisco.

J.C.J.M. Vrakking
Is coördinator op de afdeling Informatievoorziening bij de Stigas GUO Groep en projectleider voor het selectie- en invoeringstraject van een CASE-hulpmiddel.

zoek zijn voor een EDP-auditor. Buiten de algemene ontwikkel- en onderhoudsprocedures kunnen hulpmiddelen een aantal zaken afdwingen. Voorbeelden hiervan zijn:

- *Configuration management.* Met de faciliteiten voor bibliotheken- en repositorybeheer kan de betrouwbaarheid van de metagegevens beter worden gewaarborgd. Alle ontwikkelde objecten zijn hierin opgenomen, waarbij de CASE-omgeving de integriteit bewaakt.
- *Change management.* Dit omdat het bijhouden van de juiste versies in de verschillende omgevingen.
- *Autorisaties.* De omgevingen zijn dusdanig af te schermten dat er een scheiding is van ontwikkel- en gebruikersorganisatie. Binnen de ontwikkelorganisatie is dit weer te verfijnen naar rechten tot de repository. De autorisaties kunnen gelden voor bepaalde functies, voor specifieke projecten/applicaties of voor afgebakende gegevensgroepen. Het efficiënt kunnen onderhouden en het inzichtelijk maken van deze autorisaties is van belang om ze toepasbaar te houden.
- *Back-up en recovery.* De CASE-omgeving wordt dusdanig frequent gebruikt dat de reguliere backup- en recoveryprocedure met dagelijkse kopieën veelal niet voldoet. Een automatisch herstellende database kan garanderen dat geen mutaties gedurende de dag verloren gaan.
- *Documentatie.* Het traditionele knelpunt van onvoldoende of niet actuele systeemdocumentatie wordt bij CASE voor een belangrijk deel opgeheven. Het is nu eerder een kwestie van selectief de juiste rapportages gebruiken. Naast systeemdocumentatie kunnen IPSE-producten ook projectgegevens genereren om de voortgang en inzet van mensen en middelen vast te stellen.

Ontwikkelde applicaties

Voor het kunnen doen van uitspraken over de mate

van betrouwbaarheid en controleerbaarheid van de gegevensverwerking door een transactiesysteem kan een systeemaudit worden gebruikt. Deze systeemaudit gaat dan in op de eindproducten uit de CASE-omgeving, namelijk de applicaties met bijbehorende documentatie. In deze audits komen aspecten aan bod waarop de CASE-omgeving invloed heeft. Dit betreft de aanwezigheid van met name geprogrammeerde controles, zoals is te realiseren met onder meer 'triggers', audit trails, referentiële integriteitscontroles en 'constraints'. Ook de betrouwbaarheid van koppelingen, zowel tussen interne als met externe systemen (EDI), is met een goed repositorygebruik af te dwingen.

Enkele CASE-omgevingen hebben de mogelijkheid de in de applicatiemenu's of het databasemanagementsysteem opgenomen autorisatiestructuur te specificeren. Hiermee is de organisatorische processen functie-indeling te vertalen in autorisaties tot de ontwikkelde applicatie.

De repository is niet alleen een onontbeerlijke faciliteit voor teams van systeemontwikkelaars en metagegevensbeheerders, maar is ook een uitstekende bron voor EDP-auditors. Met een repository is een goed inhoudelijk inzicht in de applicatieportfolio te verkrijgen en zijn verschillende controles uit te voeren⁸. In het geval van een dynamische repository is hieruit af te leiden welke geprogrammeerde controles op dat moment gelden voor de onderzochte applicaties. Dit houdt in dat bij dynamische repository's, die zich noodzakelijkerwijs in de productieomgeving moeten bevinden, het autorisatieaspect belangrijk is.

De rol van de repository zal voor EDP-auditors aan belang gaan winnen doordat deze steeds meer de centrale spil van de ontwikkelomgeving vormt.

Dankwoord

Dit artikel is totstandgekomen in samenwerking met mevrouw A.G. Scholtalbers van TVM Verzekeringen.

⁸ In feite is sprake van een Computer Aided Audit Technique (CAAT) voor een ontwikkelomgeving.

Benchmarking van systeemontwikkeling

Drs. J.C. de Boer, ir. J.A.M. Donkers RE en ir. K.M. Lof

Hoe beheersen andere organisaties het systeemontwikkelingsproces? Zijn wij als organisatie bijzonder als we niet over een systeemontwikkelingsmethodiek beschikken of geen moderne hulpmiddelen toepassen? In welke mate betrekken we de gebruikers bij de systeemontwikkeling en hoe is het projectmanagement ingericht ten opzichte van andere organisaties? Dit artikel beschrijft de uitkomsten van benchmarkonderzoeken die als doel hadden een antwoord te geven op deze vragen.

INLEIDING

Met het toenemende belang van informatietechnologie in organisaties worden er steeds hogere eisen gesteld aan de kwaliteit van de informatiesystemen en de wijze waarop deze binnen de systeemontwikkelingsorganisatie totstandkomen.

Vanzelfsprekend bestaan bij organisaties vragen over: Hoe beheersen andere organisaties het systeemontwikkelingsproces? Zijn wij bijzonder als we niet over een systeemontwikkelingsmethodiek beschikken of geen moderne hulpmiddelen toepassen? In welke mate betrekken we de gebruikers bij de systeemontwikkeling en hoe is het projectmanagement ingericht ten opzichte van andere organisaties?

Veelal kan er geen antwoord worden gegeven op de bovenstaande vragen. Immers, gegevens van andere organisaties zijn vaak niet beschikbaar, dan wel zijn niet objectief of te gefragmenteerd. De achterliggende gedachte bij het beantwoorden van de vragen komt voort uit het willen weten wat de positie is ten opzichte van de concurrenten.

In dit artikel worden benchmarks gegeven waarmee de lezer deze vragen kan beantwoorden.

ACHTERGROND VAN DE UITGEVOERDE BENCHMARKONDERZOEKEN

In de benchmarkonderzoeken wordt gekeken naar een aantal beheersgebieden. Ten behoeve van dit artikel zijn de volgende gebieden uit de bredere benchmarkvragen gedestilleerd:

1. *IT-planning*. Formuleert de organisatie langere termijnplannen voor de ontwikkeling van informatiesystemen?
2. *managementbetrokkenheid*. Op welke wijze wordt het management betrokken bij de ontwikkeling van informatiesystemen?
3. *gebruikersbetrokkenheid*. Op welke wijze worden de gebruikers betrokken bij de ontwikkeling van informatiesystemen?
4. *projectmanagement*. Hoe wordt het project gestuurd en beheerst?
5. *systeemontwikkelingsmethodiek*. Hanteert de organisatie methodieken voor de ontwikkeling van systemen?
6. *documentatie*. Wordt het ontwikkelde systeem gedocumenteerd?

De data die als uitgangspunt voor dit artikel zijn genomen, komen voort uit 634 internationale benchmarkonderzoeken die de afgelopen drie jaar zijn uitgevoerd. Deze zijn door EDP-auditors van KPMG uitgevoerd om de volwassenheid van de IT-organisatie en de mate waarin de general IT controls worden beheerst, te beoordelen. De onderzoeken zijn uitgevoerd bij bedrijven uit de volgende branches: banken, verzekeringen, dienstverlening, handel, overheid, transport, olie en metalen, chemische industrie, elektronica, technische industrie, foodindustrie en bouw. In de benchmarkonderzoeken zijn de organisaties onderscheiden op basis van omzet en op basis van grootte van de IT-organisatie. De gehanteerde klassen van de omzet zijn: 0 tot 1 miljoen, 2 tot 10 miljoen, 11 tot 100 miljoen, 101 tot 500 miljoen, en groter dan 500 miljoen dollar. De grootte van de IT-organisatie wordt bepaald aan de hand van het aantal medewerkers. De gehanteerde klassen zijn hierbij: 1 tot 5, 6 tot 26, 26 tot 100, 101 tot 500 en meer dan 500 medewerkers.

In de benchmarkonderzoeken zijn deze aandachtsgebieden onderzocht door vragen te stellen naar het gebruik van beheersmaatregelen. De beantwoording van de vragen is gebaseerd op een groei-model ([Donk 96]). Het groei-model identificeert vijf afzonderlijke fasen. Op deze vragen kan een score worden gegeven van 1 tot en met 5. De score op de vragen wordt geverifieerd door de onderzoeker. De verificatie geschiedt door te vragen naar bewijzen voor de gegeven antwoorden. De achterliggende gedachte van de score 1 tot en met 5 is:

1. de organisatie heeft geen maatregelen getroffen;
2. de organisatie heeft informele maatregelen getroffen;
3. de organisatie heeft formele maatregelen getroffen;
4. de organisatie beheerst de maatregelen en stuurt;
5. er is een evaluatiecyclus om de maatregelen bij te stellen en af te stemmen op de wensen uit de organisatie.

In het vervolg van het artikel worden de afzonderlijke onderwerpen behandeld. Hierbij worden de

bevindingen van de benchmarkonderzoeken en de daaruit voortvloeiende overwegingen en conclusies uiteengezet.

IT-PLANNING EN -BELEID

Organisaties ontwikkelen een langetermijnvisie ten aanzien van de inzet van informatietechnologie. Het doel van deze langetermijnvisie is het waarborgen dat de juiste investeringen worden gedaan, de juiste prioriteiten worden gesteld en de juiste keuzen worden gemaakt. De visie die het management van een organisatie heeft op informatievoorziening en de wijze waarop deze moet worden bereikt, wordt vastgelegd in het informatiebeleid en/of informatieplan. In een dergelijk beleid of plan wordt onder andere aandacht besteed aan: de relatie tussen informatietechnologie en het ondernemingsbeleid, de huidige informatiesystemen inclusief de technische infrastructuur, de knelpunten in de informatievoorziening, de gewenste situatie en de wijze waarop deze situatie kan worden bereikt.

In de benchmarkonderzoeken is gekeken naar de mate van volwassenheid van organisaties ten aanzien van IT-planning en -beleid. Tabel 1 geeft een overzicht in de verdeling in scores die voor dit onderdeel zijn aangetroffen (de percentages zijn in dit artikel afgerond op gehele getallen).

Score	Percentage	Cum. percentage
1	24%	24%
2	24%	48%
3	22%	70%
4	29%	98%
5	2%	100%
Totaal	100%	

Tabel 1. IT-planning en -beleid.

Wat zegt deze tabel over de volwassenheid van IT-planning en -beleid? Van de onderzochte organisaties heeft 24% een 1 als score aangegeven. Dit wil zeggen dat deze organisaties geen aandacht hebben voor IT-planning en -beleid. Er is geen informatiebeleid en/of -plan geformuleerd. Onze ervaring leert dat deze organisaties veelal ad-hocbeslissingen nemen ten aanzien van de inzet van informatietechnologie. Investerings worden gedaan om knelpunten op te lossen zonder te kijken naar de lange termijn. Fenzelfde percentage (24%) van de organisaties scoort een 2. Bij deze organisaties is er sprake van een informeel beleid voor veelal de korte termijn. Er is geen informatiebeleid/plan op schrift gesteld; het beleid/plan zit in de hoofden van enkele personen en is vaak alleen bekend bij het automatiseringspersoneel en bij het management.

Bijna de helft (48%) van de onderzochte organisaties heeft nauwelijks of geen beleid geformuleerd of geformaliseerde plannen vastgesteld ten aanzien van de inzet van informatietechnologie.

Van de onderzochte organisaties beschikt 22% over een formeel IT-beleid en -plan (score 3). Opvallend is dat bij deze organisaties het informatiebeleid/plan voor het grootste gedeelte zonder betrokkenheid van gebruikers door de automatiseringsafdeling wordt opgesteld. Organisaties die een 4 scoren (29%) beschikken over een formeel informatiebeleid/plan dat onderwerpen bevat over onder meer de hardware, software, netwerken, automatiseringspersoneel, gebruikers, veranderingsmanagement en managementcontroles. Het opstellen van het IT-beleid en -plan gebeurt samen met alle bij de automatisering betrokken partijen. Dat wil zeggen: zowel automatiseerders als gebruikers en het (top)management. Slechts bij 2% (score 5) is IT-planning en -beleid daadwerkelijk afgeleid van het informatiebeleid en wordt daardoor gezien als een belangrijke factor voor het bijdragen aan het succes van de onderneming. Dus slechts 2% van de onderzochte organisaties werkt voor wat betreft IT-planning en -beleid volgens het boekje. Dit betekent dat in het merendeel van de organisaties informatietechnologie niet beleidsmatig wordt aangestuurd. Iets wat ongetwijfeld invloed heeft op de effectiviteit en efficiëntie van de inzet van informatietechnologie binnen organisaties.

Heeft de omvang van het automatiseringspersoneel waarover een organisatie beschikt of de grootte van de omzet nog invloed op de verdeling van de scores? Voor organisaties met een kleine IT-afdeling (tot 25 medewerkers) blijkt de verdeling niet significant te veranderen. Bij organisaties met een grote IT-afdeling (vanaf 25 medewerkers) blijkt dat er meer organisaties zijn met een formeler dan met een informeler informatiebeleid/plan (86% scoort een 3 of hoger). Zoals verwacht mag worden, neemt met het toenemen van de omvang van de IT-afdeling de behoefte aan het formeler inrichten van IT-planning en -beleid toe.

Wordt een onderscheid gemaakt op basis van de omzet, dan is eenzelfde patroon te zien. Hoe groter de omzet, des te meer aandacht er is voor het formeel inrichten van IT-planning en -beleid. Organisaties met een omzet groter dan 500 miljoen dollar beschikken nagenoeg allemaal over een formeel informatiebeleid/plan. Zij hebben daar weliswaar de tijd en ruimte voor, maar geven ook daadwerkelijk het goede voorbeeld.

MANAGEMENTBETROKKENHEID

Het onderwerp managementbetrokkenheid omvat de wijze waarop het management betrokken is bij de ontwikkeling van informatiesystemen. Uit veel publicaties valt op te maken dat informatietechnologie steeds belangrijker wordt bij het bereiken van de bedrijfsdoelstellingen. Zij wordt daarmee meer en meer een onderwerp dat voortdurend op de agenda van het (top)management moet staan.

Managementbetrokkenheid is onderzocht door te kijken naar vormen van deelname van managementlagen in werk- of stuurgroepen, de wijze van cultuur van werkgroepen en de activiteiten die worden uitgevoerd door het management. Tabel 2 geeft een

Score	Percentage	Cum. percentage
1	31%	31%
2	17%	48%
3	21%	69%
4	20%	89%
5	11%	100%
Totaal		100%

Tabel 2. Managementbetrokkenheid.

overzicht in de verdeling van scores die op dit punt door ons zijn aangetroffen.

Van de onderzochte organisaties geeft 31% (score 1) aan dat het management niet betrokken wordt bij IT-zaken. 17% (score 2) geeft aan over een stuurgroep/overlegorgaan te beschikken die op een informele manier communiceert. De communicatie betreft alleen kortetermijnactiviteiten.

Geconcludeerd kan worden dat de helft van de organisaties een 1 of een 2 scoort. Anders gezegd, bij de helft van de organisaties is het management niet of op een informele wijze betrokken bij IT-vraagstukken. Informatietechnologie is daarmee veelal nog steeds een onderwerp voor de automatiseringsafdeling en niet van de gebruikersorganisatie.

Een verdere analyse laat zien dat bij 21% (score 3) van de organisaties het management formeel betrokken is. Bij 31% (score 4 en 5) mondt de managementbetrokkenheid uit in een leidende rol voor de gebruikersorganisatie. De gebruikers en niet de automatiseerders hebben een leidende rol ten aanzien van informatietechnologie en de toepassing daarvan.

Heeft de omvang van het automatiseringspersoneel waarover een organisatie beschikt of de grootte van de omzet nog invloed op de verdeling van de scores? Uit nadere bestudering blijkt de verdeling over de scores bij organisaties met kleinere IT-afdelingen (tot 25 medewerkers) gelijk te zijn. Bij organisaties met grotere IT-afdelingen (vanaf 25 medewerkers) blijkt het management sterk betrokken te zijn bij IT-vraagstukken. Tevens blijkt de rol van de gebruikers groter te zijn dan bij organisaties met een kleinere IT-afdeling. Hetzelfde patroon is te herkennen als het onderscheid wordt gemaakt naar de omzet van de organisaties. Onze ervaring is dat bij de grotere organisaties waar informatietechnologie breed wordt ingezet, de invloed van management steeds groter wordt.

GEbruikersBETROKKENHEID

Gebruikersbetrokkenheid is een belangrijke voorwaarde voor het succesvol uitvoeren van een systeemontwikkelingsproces. Immers, het voldoende en juist betrekken van gebruikers in het ontwikkelproces draagt ertoe bij dat opgeleverde systemen voldoen aan de, soms niet expliciete, specificaties van de gebruikers. Tevens vergroot het de acceptatie van en de tevredenheid met het uiteindelijk opgeleverde systeem. Betrokkenheid van de gebruikers

wordt veelal bereikt door gebruikers gedurende het gehele ontwikkelproces actief deel te laten nemen.

Gebruikersbetrokkenheid is onderzocht door te kijken naar de mate van participatie van de gebruiker in afzonderlijke fasen van de ontwikkeling, de verantwoordelijkheden die de gebruikers hebben en de wijze van samenwerking tussen de gebruiker en de IT-organisatie. Tabel 3 geeft een overzicht in de verdeling van scores die op dit punt door ons zijn aangetroffen:

Tabel 3. Gebruikersbetrokkenheid.

Score	Percentage	Cum. percentage
1	10%	10%
2	29%	39%
3	16%	55%
4	35%	90%
5	10%	100%
<hr/>		
Totaal	100%	

De gebruikers worden in 10% van de organisaties marginaal of helemaal niet bij systeemontwikkelingsprojecten betrokken. Bij 29% (score 2) worden de gebruikers beperkt betrokken en dan nog met name tijdens de initiatiefase. De bevoegdheid tot en de personele keuze bij het betrekken van gebruikers bij een ontwikkelproject liggen dan vrijwel volledig bij de afdeling Automatisering.

Het betrekken van gebruikers in het ontwikkelproces draagt bij tot betere systemen en een hogere acceptatie. Desondanks kan worden geconcludeerd dat 39% van de organisaties nog steeds gebruikers niet intensief betreft in het ontwikkelproces.

Gebruikers betrekken bij de besluitvorming over en acceptatie van systemen vindt in 16% (score 3) van de organisaties plaats. Bij deze organisaties is het formeel vereist dat gebruikers de specificaties accorderen. Van een continue betrokkenheid gedurende het gehele ontwikkelproces is sprake bij 45% (score 4 en 5). De gebruikers worden doorlopend betrokken bij onder andere specificatie, gebruikerstests, overdracht en postimplementatie-evaluaties. De gebruikers spelen een belangrijke rol bij het aandragen en beslissen over de uiteindelijke oplossing. Opgemerkt dient te worden dat bij 35% (score 4) het project nog steeds door de IT-organisatie wordt gestuurd. Bij 10% (score 5) van de organisaties kan worden gesproken over een partnership tussen de IT-afdeling en de gebruikersorganisatie. Dit betekent onder meer dat er een klant/leverancierrelatie is gecreëerd en gebruikers en automatiseerders in samenwerking in projecten deelnemen.

Bij organisaties met een omvangrijke IT-afdeling (meer dan 25 medewerkers) blijkt bijna 80% de gebruikers doorlopend te betrekken tijdens het gehele ontwikkeltraject. Eenzelfde patroon zien wij ten aanzien van de omzet. Onze ervaring leert dat in grotere organisaties gebruikers vrijwel altijd in het verleden één of meer keren zijn geconfronteerd met applicaties die niet voldoen en die zonder tussenkomst van gebruikers zijn ontwikkeld. Als gevolg hiervan dwingen dergelijke gebruikers de participatie voor volgende projecten af.

PROJECTMANAGEMENT

De uitloop van systeemontwikkelingsprojecten in tijd en geld kan voor een groot deel worden verklaard door onvoldoende aandacht voor het projectmanagement. In de benchmarkonderzoeken is onder meer aandacht besteed aan: de projectorganisatie, de mate waarin een project wordt omschreven en afgebakend, de wijze waarop de op te leveren producten en mijlpalen worden vastgelegd in een plan van aanpak, de planning en de monitoring van de activiteiten. Tabel 4 geeft een overzicht in de verdeling van scores die op dit punt door ons zijn aangetroffen.

Score	Percentage	Cum. percentage
1	21%	21%
2	43%	64%
3	26%	90%
4	6%	96%
5	4%	100%
<hr/>		
Totaal	100%	

Tabel 4. Projectmanagement.

Projectmanagement heeft in 64% (score 1 en 2) van de organisaties geen of slechts informeel aandacht. Gezien het grote aantal systeemontwikkelingsprojecten dat een uitloop kent in tijd en/of geld, is deze uitkomst zeer opmerkelijk. Immers, systeemontwikkelingsprojecten worden steeds complexer als gevolg van bedrijfskundige en technologische ontwikkelingen. Naast het ontwikkelen van een informatiesysteem zijn hedendaagse systeemontwikkelingstrajecten steeds vaker een combinatie van:

- (her)ontwerpen en inrichten van de bedrijfsprocessen en de bijbehorende (administratieve) organisatie;
- ontwerpen, realiseren en implementeren van informatiesystemen en de daarbij behorende infrastructuur;
- (her)inrichten van de personele organisatie en het begeleiden van het totale veranderingsproces.

De noodzaak van een adequaat projectmanagement wordt daarmee alleen maar groter. Naar onze mening ligt hier voor de in het onderzoek betrokken organisaties een duidelijk aandachtspunt.

Een verdere analyse laat zien dat 26% (score 3) formele maatregelen heeft getroffen voor de beheersing van het project. Bij elk nieuw project worden een plan van aanpak en een impactanalyse opgesteld. Het project wordt opgedeeld in delen waarbij de mijlpalen worden gedefinieerd en er is sprake van formeel risicomangement. De voortgang van de activiteiten wordt vastgelegd en bewaakt. Het projectteam rapporteert over de voortgang aan het management.

10% (score 4 en 5) gaat nog een stap verder met de inrichting van het projectmanagement. Bij deze organisaties wordt de situatieanalyse van de onderneming gebruikt voor het vaststellen van de prioriteiten. De prioriteitstelling wordt gebruikt voor de ver-

deling van het automatiseringsbudget (IT-budget) over de verschillende projecten. Tevens wordt er een haalbaarheidsstudie uitgevoerd voordat met de uitvoering van een project wordt gestart. Er wordt gebruikgemaakt van formele kwaliteitsplannen en projectstandaarden. Bij 4% van deze groep (score 5) worden de projecten na afronding op een formele wijze beoordeeld. Op basis van de uitkomsten hiervan wordt het projectmanagement verder verbeterd.

Organisaties met een kleine IT-afdeling en een lage omzet hebben relatief weinig aandacht voor projectmanagement. Voor een deel kan dit worden verklaard doordat systeemontwikkeling in deze organisaties beperkt plaatsvindt en de projecten relatief klein zijn. Daarnaast bestaat bij ons de indruk dat het belang van projectmanagement in dit soort organisaties nog steeds wordt onderschat. Projectmanagement wordt veelal gezien als iets wat er naast de reguliere projectactiviteiten wordt bij gedaan. Projectmanagementactiviteiten worden dan ook vaak niet ingepland. Bij grotere organisaties treffen wij formelere vormen van projectmanagement aan.

SYSTEEMONTWIKKELINGSMETHODIEK

Het onderwerp systeemontwikkelingsmethodiek omvat de methodiek die wordt gebruikt om de ontwikkeling en het onderhoud van systemen te structureren. Voorbeelden van systeemontwikkelingsmethodieken zijn SDM-II, BSP/ISP en Oracle Custom Development Method. In de benchmarkonderzoeken is het onderwerp systeemontwikkelingsmethodiek onderzocht op de volgende kenmerken: aanwezigheid methodiek, keuze methodiek en ondersteuning van methodiek door middel van tools. Tabel 5 geeft een overzicht in de verdeling van scores die op dit punt door ons zijn aangetroffen.

In 68% (score 1 en 2) van de gevallen beschikken de organisaties of niet of over een informele systeemontwikkelingsmethodiek. In het laatste geval heeft de organisatie een aantal maatregelen genomen. Er zijn enkele tools en handleidingen voor de ondersteuning van bepaalde fasen van de levenscyclus, zoals de haalbaarheidsstudie, ontwerp en testen van het informatiesysteem. 20% (score 3) past een formele methodiek toe die alle fasen van de levenscyclus omvat en waarvan de verschillende onderdelen zijn gedocumenteerd. Het personeel van deze bedrijven is getraind in de toepassing van de methode. De methodiek wordt in de organisatie volledig door tools ondersteund. Verder wijst het onderzoek uit dat 12% (score 4 en 5) tevens over een Quality Assurance-functie beschikt die controleert of de methodiek wordt toegepast en juist wordt gehanteerd. De tools en de methodiek zijn volledig geïntegreerd en beschikken over functionaliteit om informatie over te zetten naar een volgende fase van de levenscyclus. De methodiek evolueert mee met de veranderende behoeften van het bedrijf.

De hoogte van de omzet heeft geen invloed op de verdeling in scores. Als het onderscheid tussen de

Score	Percentage	Cum. percentage
1	32%	32%
2	36%	68%
3	20%	88%
4	8%	96%
5	4%	100%
<hr/>		
Totaal	100%	

Tabel 5. Systeemontwikkelingsmethodiek.

organisaties wordt gemaakt op omvang van de IT-afdeling dan is er wel een verschil te constateren. 80% van de kleinere IT-organisaties (minder dan 25 medewerkers) scoort tussen beschikt niet of beschikt over een informele methodiek. Bij grotere IT-afdelingen (25 en meer medewerkers) scoort 70% een 2 of een 3. Zoals verwacht geeft de omvang van het aantal IT-medewerkers aanleiding voor het verder hanteren van een ontwikkelmethodiek.

DOCUMENTATIE

Tot slot wordt ingegaan op het onderwerp documentatie. Goede documentatie draagt ertoe bij dat exploitatie en onderhoud voor de IT-afdeling worden vereenvoudigd. Tevens draagt zij bij aan het effectief en efficiënt gebruik van het informatiesysteem door gebruikers. In de benchmarkonderzoeken is onder meer onderzocht: de beschikbaarheid, de actualiteit en de samenhang tussen de verschillende soorten documentatie.

Score	Percentage	Cum. percentage
1	10%	10%
2	27%	37%
3	48%	85%
4	11%	96%
5	4%	100%
<hr/>		
Totaal	100%	

Tabel 6. Documentatie.

Er is geen of weinig documentatie beschikbaar in 10% (score 1) van de organisaties. Bij 27% (score 2) is er wel documentatie aanwezig maar deze is niet up-to-date. De documentatie komt niet geheel overeen met de huidige systeemspecificaties. Veranderingen in het systeem worden niet op tijd bijgewerkt in de documentatie. 48% (score 3) beschikt over een goedkeuringsproces dat verzekert dat systeemwijzigingen in de documentatie worden bijgewerkt. Echter, er is geen coördinatie over de verschillende documentatie zodat volledige consistentie niet is gewaarborgd. Documentatie is beschikbaar voor iedereen die het nodig heeft. Verder blijkt dat 15% (score 4 en 5) de documentatie periodiek beoordeelt. Tevens worden bij deze organisaties projecten niet goedgekeurd voordat de documentatie compleet is. Coördinatie verzekert de consistentie tussen verschillende soorten documentatie.

Geconcludeerd kan worden dat 37% van de bedrijven niet over of over niet-actuele documentatie be-

Drs. J.C. de Boer
Is als EDP-auditor werkzaam bij KPMG EDP Auditors in Utrecht. Zijn aandachtsgebied ligt op het gebied van advies en audits ten aanzien van management van IT. Daarnaast is hij Nederlands projectleider IT Controls Benchmarking.

schikt. Uit ervaringen bestond de verwachting dat dit percentage hoger zou moeten zijn. Een verklaring voor het voor ons gevoel lage percentage kan mede worden gevonden in het feit dat organisaties in de beoordeling ook de documentatie van standaardpakketten meenemen. Deze documentatie (al dan niet elektronisch) is veelal uitgebreider en heeft een hogere actualiteitswaarde.

De omvang van de IT-afdeling heeft geen significante invloed op de verdeling. Als er een onderscheid wordt gemaakt op basis van de omzet dan is er wel een verschil aan te treffen. In tegenstelling tot wat mag worden verwacht, blijkt dat organisaties met een lagere omzet op het punt van documentatie hoger scoren dan die met een hogere omzet. Dit verschil kan naar onze mening worden verklaard doordat kleine organisaties minder aan systeemontwikkeling doen en veelal beschikken over standaardpakketten die relatief goed zijn gedocumenteerd.

CONCLUSIE

In het artikel zijn achtereenvolgens zes aandachtsgebieden van systeemontwikkeling aan de orde gekomen. Voor elk van deze gebieden is er veel aandacht binnen de IT-wereld. IT-planning en -beleid zijn belangrijke voorwaarden voor het succesvol toepassen van informatietechnologie, het management en de gebruikers moeten worden betrokken bij IT-vraagstukken en projectmanagement is een voorwaarde om systeemontwikkelingsprojecten te beheersen. Hetzelfde geldt voor het toepassen van een systeemontwikkelingsmethodiek en het documenteren van informatiesystemen.

Kijken we naar de omvang van de IT-afdeling en de grootte van de omzet dan zien we een beeld dat in de lijn der verwachting ligt. Hoe groter de IT-afdeling en de omzet des te meer aandacht er is voor het nemen van beheersmaatregelen ten aanzien van de zes aandachtsgebieden. De conclusie zou kunnen worden getrokken dat grotere organisaties volwassen zijn op het gebied van systeemontwikkeling dan kleinere organisaties.

Wordt het onderscheid gemaakt naar grootte van de IT-organisatie (het aantal medewerkers), dan ziet men dat de scores normaal verdeeld zijn rond score 2 voor organisaties met een IT-organisatie tot 5 medewerkers. Bij IT-organisaties van 6 tot 25 medewerkers ziet men een verdeling rond score 3. Grote IT-organisaties, vanaf 25 medewerkers, hebben hoofdzakelijk geformaliseerde beheersmaatregelen.

Onze belangrijkste conclusie is dat de praktijk weerbarstiger is dan het ideaalbeeld dat vaak wordt gehanteerd. In de praktijk blijkt 52% van de organisaties geen of slechts informele beheersmaatregelen te hebben genomen ten aanzien van de zes aandachtsgebieden. De conclusie zou kunnen worden getrokken dat het merendeel van de organisaties nog niet volwassen is op het gebied van systeemontwikkeling.

Bij veel van onze cliënten treffen wij zeer ambitieuze systeemontwikkelplannen aan. Indien we dit combineren met onze belangrijkste conclusie inzake de beperkt aanwezige beheersmaatregelen, is dat aanleiding tot zorg.

Wij adviseren in het algemeen organisaties met dergelijke ambitieuze ontwikkelplannen de bestaande beheersmaatregelen in kaart te brengen en te verbeteren. Ze vormen immers een onontbeerlijke basis voor het uitvoeren van dergelijke projecten.

LITERATUUR

[Donk96] J.A.M. Donkers en E.R. van Sommeren, *Benchmarking, een hulpmiddel voor de EDP-auditor?*, Compact 1996/2.

[Somm97] E.R. van Sommeren, J.C. de Boer en J.A.M. Donkers, *IT Benchmarking*, Automatisering Gids, oktober 1997.

[Vlis97] A. van der Vlist en P. Noordam, *Trends in IT*, Op tijd investeren in de juiste technologie, Kluwer Bedrijfsinformatie, 1997.

Multimedia in open omgevingen

De mogelijkheden van het World Wide Web

Drs. A.M. Buren

In toenemende mate worden bedrijfssystemen door websites gekoppeld aan het World Wide Web. Het benutten van het Web als medium voor electronic commerce-toepassingen is daarbij van velerlei factoren afhankelijk. In dit artikel worden de mogelijkheden, ontwikkelingen en auditaspecten besproken die bepalend zijn voor het succes van dergelijke toepassingen. Door tijdens de systeemontwikkeling hieraan invulling te geven kunnen de commerciële mogelijkheden van het Web zo goed mogelijk worden benut.

INLEIDING

De laatste jaren is geen superlatief ongebruikt gelaten om de mogelijkheden van nieuwe media als Internet en intranet aan te geven. Onder de verzamelnaam *electronic commerce* zijn tal van nieuwe informatietoepassingen reeds ontwikkeld of nog in ontwikkeling. Het Internet vormt een nieuw communicatiekanaal voor het aanbieden van producten, diensten en informatie, met in opzet een mondiaal bereik. Maar ook intranet biedt veelbelovende toepassingen, waarbij binnen een organisatie door effectievere interne berichtgeving de productiviteit van medewerkers kan worden verbeterd.

De populariteit van deze nieuwe media is vooral toegenomen door het grafisch georiënteerde World Wide Web, ook wel kortweg het Web genoemd. In toenemende mate worden transactiesystemen gekoppeld aan het Web. Hierbij vormen de zoek- en beheerfaciliteiten van transactiesystemen samen met de laagdrempelige, gebruikersvriendelijke interface van de web-omgeving, een krachtige combinatie voor nieuwe informatietoepassingen genaamd websites of web-media.

Web-media onderscheiden zich van conventionele transactiesystemen door het gebruik van andere op Internet-technologie gebaseerde communicatie- en presentatietechnieken. Door het snelle tempo waarin deze technologieën zich ontwikkelen worden de mogelijkheden van web-media steeds geavanceerder. Naast statische informatieoverdracht bieden deze technologieën nu mogelijkheden tot verhoogde interactiviteit in de informatieoverdracht door het dynamisch genereren van web-pagina's. Met gebruikmaking van de programmeertaal Java kan daarbij de informatieoverdracht zelfs onafhankelijk van het onderliggende hardwareplatform worden gemaakt.

Hoewel de technologische levenscyclus van web-media zich nog in een ontwikkelingsstadium bevindt, waarbij huidige implementaties van web-media vaak geïnitieerd zijn door prestigieuze overwegingen, begint ook hier het besef door te dringen dat *web-media nimmer een doel op zich mogen vormen, maar op effectieve wijze ondersteuning dienen te bieden aan bedrijfsprocessen*. Web-media dienen door een kwalitatieve opzet voor gebruikers meerwaarde te bieden ten opzichte van andere vormen van informatieverspreiding.

In dit artikel worden de factoren geanalyseerd die bepalend zijn voor het succes van web-media. Hiertoe zal het artikel ingaan op de volgende vragen:

- Wat zijn web-media en hoe zijn deze ontstaan?
- Waarom web-media? De meerwaarde van web-media wordt geanalyseerd op gebruikers- en organisatieniveau.
- Waarmee zijn web-media mogelijk geworden? De bouwstenen van web-technologie worden nader besproken.
- Waarheen? Afsluitend worden belangrijke toekomstige ontwikkelingen belicht.

Het tijdens de systeemontwikkeling van web-toepassingen uitwerking geven aan de hierbij gesignaleerde mogelijkheden, ontwikkelingen en auditaspecten zal zich vertalen naar een hogere mate van effectiviteit, efficiëntie en manageability van dergelijke toepassingen.

NIEUWE MEDIA – HET WEB

Deze paragraaf beschrijft hoe web-media als gevolg van Internet en intranet zijn ontstaan en komt tot be­gripsbepaling hiervan.

Historische ontwikkeling Internet

De verhalen over het ontstaan van het Internet mo­gen inmiddels genoegzaam bekend zijn. In de jaren zestig, ten tijde van de koude oorlog, zijn de eerste fundamente­nen gelegd voor het huidige Internet ([Paan96]). Verschillende van strategisch belang zijnde computercentra werden op zodanige wijze gekoppeld dat tijdens een (nucleaire) aanval de communicatie tussen deze computercentra behou­den zou blijven. Hiertoe werd gekozen voor een net­werk­stelsel waarbij informatieoverdracht niet via doorgaans vaste (statische) routeringsroutes verliep, maar waar kleine informatiepakketjes via variabele routeringsroutes los van elkaar worden gestuurd. De hiervoor gebruikte protocollen – voorlopers van het huidige TCP/IP-protocol – bepalen nu in be­langrijke mate de wijze waarop computersystemen binnen het Internet communiceren.

Het besturingssysteem Unix, dat eind jaren zestig werd ontwikkeld, maakte het op een relatief een­voudige wijze mogelijk computersystemen door middel van TCP/IP onderling te koppelen. Met de toename van koppelingen werd de basis gelegd voor het huidige Internet. In de jaren zeventig en tachtig werden hieraan steeds meer lokale netwer­ken toegevoegd, vooral om electronic mail te kun­nen uitwisselen. Het intensieve gebruik door de we­enschappelijke wereld heeft de verdere groei gesti­muleerd. Begin jaren negentig groeide het Internet explosief door toename van gebruikers uit de priva­te sector. De ontwikkeling van het visueel georiën­teerde World Wide Web in 1992, waardoor het Inter­net bereikbaar werd voor een breed publiek, speelde hierin een cruciale rol.

Momenteel zijn er bij benadering circa 50 miljoen gebruikers op het Internet actief via 50.000 verschil­lende netwerken. Het Internet is daarmee een net­werk van netwerken geworden.

De communicatie

Aangezien het Internet een netwerk van technisch

verschillende netwerken is, is het noodzakelijk afspraken te maken over de wijze waarop de compu­ters met elkaar dienen te communiceren. Bij de com­municatie van gegevenspakketjes wordt gebruik­gemaakt van het Internet Protocol (IP) en het Trans­mission Control Protocol (TCP)¹ voor monitoring van de volgtijdelijke juistheid van deze gegevenspak­ketjes in de overdracht. De TCP/IP-architectuur be­staat, gezien vanuit de Open Systems Interconnection (OSI)-gedachte, uit een viertal lagen van onderlig­gende protocollen, die met elkaar communiceren op basis van interfaces zoals weergegeven in figuur 1.

TCP/IP is oorspronkelijk sterk verbonden met het Unix-platform, maar langzamerhand uitgegroeid tot een de-factocommunicatiestandaard. Het TCP/IP-protocol wordt nu op nagenoeg ieder hardwareplat­form ondersteund en is daarmee een communi­catieprotocol dat onafhankelijk is van leverancier, platform, besturingssysteem en het onderliggende fysieke netwerk.

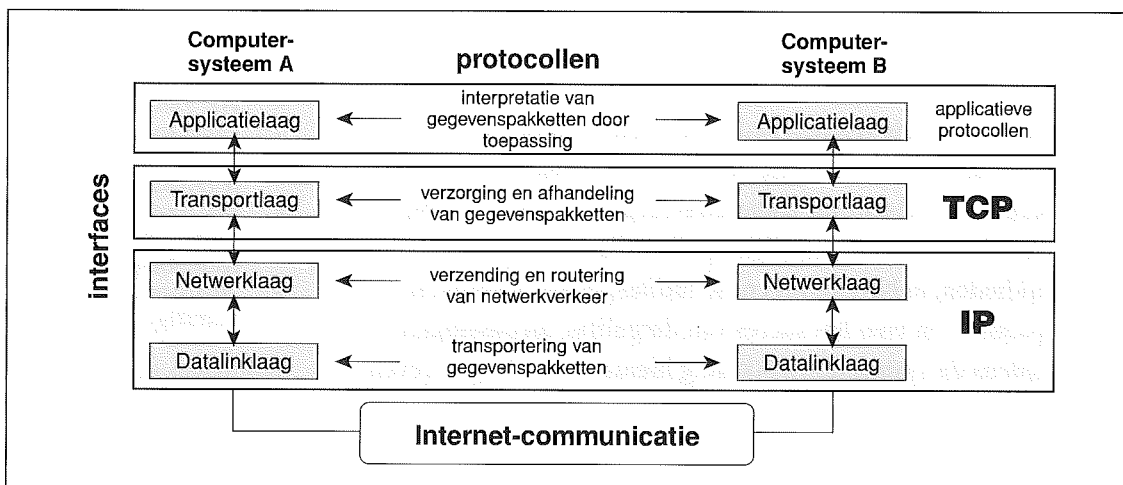
Naast afspraken over de wijze waarop de compu­ters met elkaar dienen te communiceren, is het voor de routing van de informatieoverdracht noodza­kelijk te beschikken over een uniform adressering­stelsel. Ondanks het vaak anarchistische karakter van het Internet zijn hierover duidelijke afspraken gemaakt. Elk computersysteem in het Internet heeft een uniek IP-adres (bijvoorbeeld 194.178.56.21). Aan de hand van het IP-adres vindt de routing van informatiepakketjes plaats, waardoor onderlinge communicatie mogelijk wordt. Omdat het gebruik van numerieke adressen weinig gebruiksvriendelijk is, worden IP-adressen afgebeeld op domeinnamen (zoals www.bedrijfsnaam.nl). Hiertoe wordt ge­bruikgemaakt van het hiërarchische bewegwijze­ringssysteem Domain Name System (DNS).

Met behulp van eerder beschreven architectuur van protocollen en gebruikmakend van een eenduidig adresseringssysteem zijn in de loop der jaren allerlei toepassingen op het Internet gerealiseerd, elk met eigen applicatieve protocollen. Voorbeelden hiervan zijn e-mail (electronic mail), FTP (bestandsover­dracht), Telnet (terminalemulatie), Chat, Gopher en het World Wide Web (WWW).

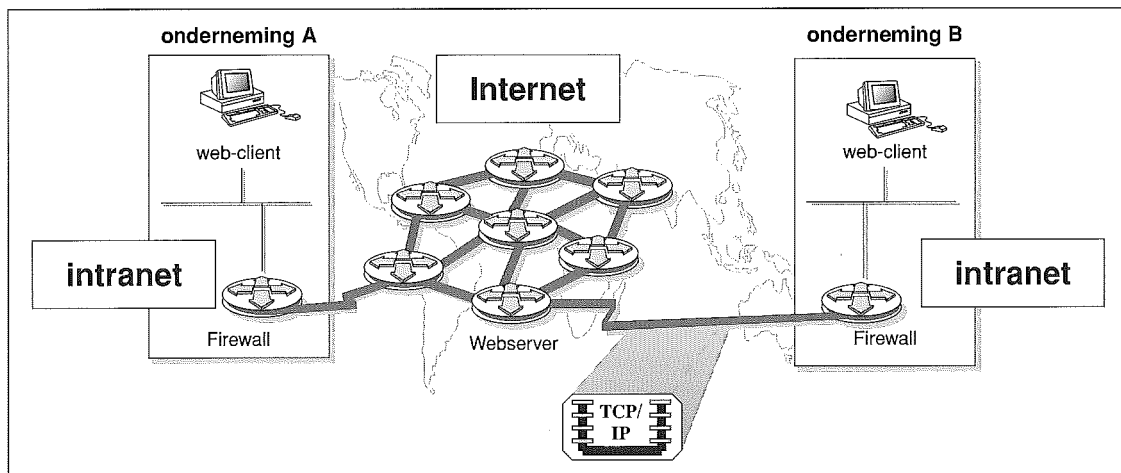
World Wide Web

De populariteit van het Internet is vooral toegenomen

1 Naast TCP wordt voor controle op gegevenspakketjes in mindere mate ook het User Datagram Protocol (UDP) gebruikt. Dit protocol is zogeheten 'connection less'-georiën­teerd, waardoor het is te ge­bruiken voor toepassingen (als e-mail) waarbij zender en ont­vanger geen sessie aangaan. Het UDP-protocol is hierdoor beperkt tot controles op de in­houd van gegevenspakketjes en voorziet niet in controles op de volgorde en volledigheid van de berichtenstroom.



Figuur 1. TCP/IP-architectuur.



Figuur 2. Internet en intranet.

men door de gebruikersvriendelijke grafische toepassing World Wide Web, kortweg het Web genoemd. Op het Web fungeren pagina's als de eenheid voor informatieoverdracht. De opmaak van deze pagina's wordt hierbij vastgelegd in de presentatiestandaard HyperText Markup Language (HTML). Deze HTML-pagina's worden daarbij, met behulp van het communicatieprotocol TCP/IP en het applicatieve protocol HyperText Transaction Protocol (HTTP), van computersysteem naar computersysteem getransporteerd.

Een HTML-pagina is gebaseerd op het principe van hyperlinks. Een hyperlink is een verbinding tussen verschillende informatieonderdelen. Door middel van hyperlinks worden HTML-pagina's op een gebruikersvriendelijke wijze met elkaar verbonden en kunnen gebruikers op een interactieve en associatieve wijze over het Web navigeren. Naast tekst kunnen HTML-pagina's ook informatie bevatten in de vorm van afbeeldingen, geluid, animatie en video-beelden.

Het gezamenlijk gebruik van meerdere informatievormen in de informatieoverdracht, in een interactieve omgeving, wordt ook wel aangeduid als multimedia. De multimediale presentatie van HTML-pagina's op het World Wide Web heeft, vooral door verhoging van de amusementswaarde van de informatieoverdracht, haar meerwaarde voor gebruikers bewezen ([Bure95]). Mede hierdoor is het gebruik van het Internet de afgelopen jaren explosief gestegen.

Het World Wide Web, het Internet en intranet worden vaak in één adem genoemd. Toch is het belangrijk deze drie goed te onderscheiden. Het World Wide Web is een toepassing waarbij pagina's aan de hand van standaarden als HTTP en HTML worden opgeslagen, gedistribueerd en gekoppeld. Internet en intranet zijn technische infrastructuren waarover al dit verkeer wordt afgehandeld met behulp van netwerkprotocollen, domeinnamen en IP-adressen.

In dit artikel staan World Wide Web-toepassingen op het Internet en intranet centraal. Deze toepassingen zullen kortweg als Web worden aangeduid. Overige toepassingen op het Internet en intranet worden buiten beschouwing gelaten.

Internet en intranet

Internet-technologieën als TCP/IP, HTML en HTTP kunnen behalve voor het publieke Internet ook worden gebruikt om informatie over een kleinere groep te verdelen, via een zogeheten intranet. Een intranet betreft een lokaal netwerk binnen de grenzen van een onderneming. Een firewall fungeert hierbij vaak als een begrenzingssysteem tussen het intranet en het Internet.

Technisch gezien is er geen of weinig verschil tussen een intranet en het Internet. De technologieën onder Internet en intranet zijn hetzelfde. Pagina's worden binnen een intranet op gelijke wijze getransporteerd als pagina's op het Internet.

Internet en intranet onderscheiden zich voornamelijk in het gebruik. Het intranet wordt veelal gebruikt als medium voor interne bedrijfsinformatie. Zo zijn het ontsluiten van interne databases en het publiceren van bedrijfsdocumenten typische intranet-toepassingen. De informatie die door een bedrijf aan het Internet wordt toevertrouwd, is anders van aard. Dan gaat het bijvoorbeeld om een profielschets van het bedrijf, een overzicht van het aanbod aan producten en diensten of om informatie bedoeld als klantenservice.

Web-media

De op het Web aan te bieden informatie is veelal opgeslagen in conventionele transactiesystemen. In toenemende mate zijn en worden deze systemen gekoppeld aan het Web, waaruit nieuwe en commercieel interessante informatietoepassingen ontstaan. Federal Express heeft hierbij de toon gezet door zijn callcenter te vervangen door een website. Klanten kunnen hierdoor met een web-browser, waar ook ter wereld, eenvoudig de status van verzonden pakketjes opvragen. Naast het concurrentievoordeel werd per raadpleging een kostenreductie van 15 naar 0,15 dollar bereikt. Andere generieke voorbeelden van gerealiseerde web-toepassingen zijn registratieformulieren, ordersystemen, search engines, news services, shopping systems, en zelfs electronic banking.

In dit artikel staan de functionele en technische aspecten van deze nieuwe toepassingen centraal. Toepassingen die door middel van media als Internet dan wel intranet in een interactieve web-omgeving informatie genereren. Interactief betekent hierbij dat

de inhoud van gegevensoverdracht door de gebruikers kan worden bepaald. Het samengaan van één medium van een informatiesysteem gebruikmakend van een web-omgeving wordt daarbij ook wel een website² genoemd. Bij de analyse hiervan wordt de volgende begripsbepaling gehanteerd:

Website: een door computer(s) gecontroleerd systeem (webserver) van generatie, manipulatie, presentatie, opslag en distributie van gegevens, waarbij de presentatie van de informatieoverdracht uit meerdere informatievormen kan bestaan, distributie over het Internet dan wel intranet plaatsvindt en interactief in een web-omgeving door een gebruiker kan worden bepaald.

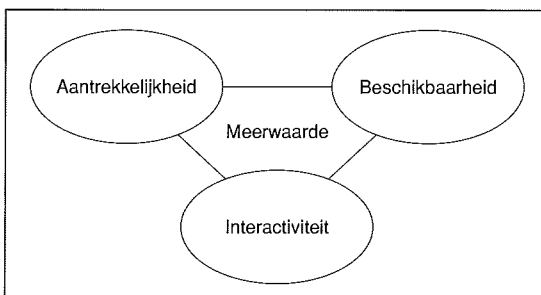
MEERWAARDE VAN WEB-MEDIA

Wat is de meerwaarde van een website? Hoe creëert en behoudt een onderneming deze meerwaarde? Anders geformuleerd: wat zijn factoren die bepalend zijn voor het succes van een website en direct of indirect tot economische baten kunnen leiden?

De inzet van het Web als communicatiekanaal zal een weloverwogen beslissing dienen te zijn. Een website is meer dan een technische toepassing; het betekent een verschuiving van het initiatief tot informatie-uitwisseling naar individuele gebruikers. De perceptie van gebruikers ten opzichte van de website bepaalt hierdoor in hoge mate het gebruik ervan. De gebruiker zal via de website door een complexe hoeveelheid onderling verbonden pagina's naar de juiste informatie dienen te worden geloodst. Bij het inrichten van web-media is het daarom van belang een kwalitatief gedegen opzet te hanteren en de succesfactoren van het Web hierin te betrekken.

De mate waarin daadwerkelijk meerwaarde wordt behaald, is daarbij afhankelijk van de invulling van deze succesfactoren. In een uit te voeren audit naar de effectiviteit van een website fungeren deze succesfactoren dan ook als onderzoeksobjecten. Daarbij is het van groot belang de meerwaarde te behouden door afdoende beheer- en onderhoudsinspanningen. In toenemende mate worden het inrichten, het beheer en het onderhouden van een website dan ook uitbesteed aan daarin gespecialiseerde organisaties.

Achtereenvolgens worden de succesfactoren en de invulling daarvan geanalyseerd op gebruikers- en organisatieniveau.



² Naast de termen *website* en *web-medium* worden hier voor ook wel benamingen als 'web-databases', 'Webbases', 'Interactive Web Services' en 'Open Gateway Technology' gehanteerd.

Figuur 3. Meerwaarde web-media op gebruikersniveau.

Meerwaarde op gebruikersniveau

Web-toepassingen zijn multimediatoepassingen. Zoals de definitie laat zien is een website een geautomatiseerd databasesysteem waarbij de inhoud van de informatieoverdracht uit meerdere informatievormen kan bestaan, over het Internet dan wel intranet wordt gedistribueerd en door een gebruiker interactief in een web-omgeving kan worden bepaald. Multimedia is een eigenschap die aangeeft dat verschillende informatievormen (tekst, graphics, geluid, beeld en video) gezamenlijk in één medium gehanteerd kunnen worden en interactie kan plaatsvinden met een gebruiker ([Bure95]). Web-media beschikken daarvoor over de multimediale eigenschap en zijn bijgevolg multimediatoepassingen.

Op gebruikersniveau beschikken multimediatoepassingen over drie succesfactoren, te weten ([Bure95]):

- *aantrekkelijkheid*, ook wel *aibaarheid* ([Dort96]) genoemd. Multimedia maakt een datavisualisatie mogelijk die – door het gebruik van meerdere informatievormen – beter is afgestemd op (zintuiglijke) behoeften van gebruikers;
- *interactiviteit*. Multimedia biedt de mogelijkheid om op de verstrekte informatie te reageren en het vervolg van het aanbod inhoudelijk te beïnvloeden;
- *beschikbaarheid*. Multimedia impliceert een effectieve beschikbaarheid van informatie door media als Internet en intranet, dan wel door opslagmedia als cd-rom en cd-i.

In opzet bestaat de meerwaarde van een website bijgevolg uit het totaal van de drie generieke succesfactoren aantrekkelijkheid, interactiviteit en beschikbaarheid, zoals weergegeven in figuur 3.

Een website zal voor de ontvanger meerwaarde moeten bieden in de vorm van tijdig beschikbare, interactieve informatie met een hoge mate van aantrekkelijkheid. De mate waarin daadwerkelijk meerwaarde wordt behaald, is daarbij afhankelijk van de invulling van de succesfactoren. Op hoe invulling te geven aan deze succesfactoren wordt hieronder ingegaan.

Aantrekkelijkheid

Een website is een sterk visueel medium. Voor de analyse van het aspect aantrekkelijkheid is onderscheid gemaakt in vormgeving en inhoud.

Vormgeving

Ter verhoging van de betrokkenheid van de gebruiker is een visueel attractieve vormgeving van groot belang ([Bure95]). De vormgeving van het eerste beeldscherm, de home-page van een website, is vaak bepalend voor het succes van de website. Voor het bewerkstelligen van een laagdrempelige, gebruikersvriendelijke inrichting is een heldere structuur noodzakelijk. Een duidelijke, consistente (grafische) vormgeving bevordert een overzichtelijke, hanteerbare en begrijpelijke structuur.

Gebruikers hebben behoefte aan voorspelbaarheid en herkenbaarheid. Herkenbaarheid is bepalend voor de identiteit van de website. Eenduidige functionaliteit en grafische continuïteit van de verschillende informatieonderdelen zijn hierbij van groot belang, met name als deze bestaan uit vele verschil-

lende onderwerpsgebieden en een omvangrijke hoeveelheid menu's en pagina's. Het hanteren van een eigen uniforme huisstijl is daarbij onontbeerlijk.

Om de herkenbaarheid te vergroten is het van belang om een eenduidige opmaak te hanteren, die steeds herkenbaar is en die min of meer onafhankelijk is van de plaats waar de gebruiker zich binnen de website bevindt. Pagina's dienen zodanig te zijn opgezet dat de informatie duidelijk overkomt en dat de gebruiker steeds weet waar hij zich bevindt. Hiervoor dient elk informatieonderdeel ter oriëntatie van de gebruiker te beschikken over een herkenbare titel, eventueel visueel vormgegeven in een afbeelding. Ook tussenkoppen en paragraafaanduidingen dienen duidelijk herkenbaar te zijn. Zaken van extra belang moeten zijn geaccentueerd. Het gebruik van scripts en applets kan hierbij ondersteuning bieden (waarover meer in de paragraaf 'Webtechnologie'). Hierbij moet worden gewaakt voor overmatig gebruik van afbeeldingen, (reclame)logo's, animaties en kleuren. Al deze grafische toevoegingen kunnen de aandacht afleiden van de inhoud.

Inhoud

Naast een aantrekkelijke vormgeving is een kwalitatief inhoudelijke vulling van de website van groot belang³ ([SUN96]). In het algemeen vertonen gebruikers van een website egocentrische en haastige gedragspatronen. Uit nieuwsgierigheidsoverwegingen wordt in de regel slechts vluchtig een bezoek gebracht aan een website. De meerwaarde van de website voor gebruikers dient zich dan al snel te openbaren, door het wekken van interesse, anders zal de website weer snel worden verlaten. De home-page moet hiervoor bij voorkeur één centrale doordachte boodschap (leidmotief) bevatten die bewerkstelligt dat alle pagina's op de website als één logisch geheel bij de gebruiker overkomen. De aangeboden informatie dient daarbij bruikbaar, zinvol en betrouwbaar te zijn.

Waar het in de praktijk doorgaans aan schort is de website aantrekkelijk te houden. Om ervoor te zorgen dat gebruikers die de website eenmaal hebben bezocht regelmatig terugkeren, is het zaak door voortdurende verversing van de aangeboden informatie de website actueel te houden. Wanneer de aangeboden informatie door de tijd is achterhaald, zullen gebruikers snel afhaken. Documenten in het algemeen, dus ook web-pagina's, hebben een levenscyclus. Hierbij dient, naast de technische verantwoordelijkheid voor het onderhoud van een website, de functionele verantwoordelijkheid (eigenaarschap van de inhoud) van pagina's goed te zijn vastgelegd.

Interactiviteit

Een website is een interactief medium. Voor de analyse van interactiviteit is onderscheid gemaakt in toegankelijkheid en betrokkenheid.

Toegankelijkheid

Informatie op een website dient toegankelijk te zijn. Hyperlinks zijn hiervoor bruikbare middelen. De kracht van het Web ligt in de grote hoeveelheid hyperlinks waarmee toegang tot pagina's en andere web-locaties mogelijk is. Een mogelijk neveneffect hiervan is dat gebruikers snel gedesoriënteerd ra-

ken, als van een duidelijk begin-, midden- en eindpunt geen sprake meer is. De kunst is om een uitgebalanceerde (hiërarchische) structuur samen te stellen, waardoor de gebruikers op een vlotte wijze de gewenste informatie kunnen vinden en waarbij ze de wijze waarop de website is ingericht snel doorzien. Er bestaan vooralsnog geen volwassen inrichtingsprincipes voor meerdimensionele elektronische informatieomgevingen die zo'n uitgebalanceerde structuur garanderen.

Daarnaast kan de toegankelijkheid worden verbeterd door het gebruik van navigatiebuttons. Navigatiebuttons dienen – boven op de standaard-navigatiebuttons in de web-browser – additionele functionaliteit te bieden om op een efficiënte wijze door de pagina's van de website te kunnen bladeren. Een ander middel ter verhoging van de interactiviteit is het bieden van zoekfaciliteiten. Hierdoor hoeft de gewenste informatie niet te worden gevonden door het aflopen van hyperlinks en navigatiebuttons, maar kunnen relevante pagina's direct aan de hand van een sleutelveld worden gepresenteerd.

De inhoud van een website moet regelmatig worden verversst.

Betrokkenheid

Door het bieden van interactieve responsystemen, zoals elektronische reactie- en evaluatieformulieren en vraagbaken met behulp van e-mail, kan de website verder worden afgestemd op de gebruikersgroep. De betrokkenheid van de doelgroep kan, mits de reacties zichtbaar tijdig worden opgevolgd, hierdoor toenemen, hetgeen de kans op terugkeer vergroot.

Ook het personaliseren van de website door het gebruik van identificatie- en begeleidingssystemen kan deze betrokkenheid vergroten. Is een gebruiker eenmaal geïdentificeerd, dan is een website met eenvoudige vormen van kunstmatige intelligentie zo in te richten dat deze meer gaat anticiperen op de keuzen van de gebruiker en zich steeds meer aan het denkproces van de desbetreffende gebruiker gaat aanpassen (zogenaamde adaptieve interactiviteit).

Beschikbaarheid

Een website impliceert een effectieve beschikbaarheid van informatie. Voor de analyse van de beschikbaarheid is onderscheid gemaakt in bereikbaarheid en snelheid.

Bereikbaarheid

Een website biedt een (grote) hoeveelheid van benaderbare gegevens via het Internet of intranet. Alvorens een gebruiker hierover kan beschikken, dient ten eerste de website op het Internet vindbaar te worden gemaakt op het Web. Een voor de hand liggende DNS-naamgeving is hierbij het startpunt. Toegang tot een website vindt veelvuldig plaats door een referentie via geactiveerde hyperlinks op andere web-media. Het is van belang de aanwezigheid van relevante referentiesites te inventariseren

3 Het hangt daarbij van het gebruikte systeem af in welke vorm(geving) de informatie de gebruiker bereikt. Dit is mede afhankelijk van de gebruikte browser en van de instellingen van het gebruikte systeem.

en waar mogelijk een hyperlink naar de eigen website te laten opnemen. De website dient daarnaast bekend te zijn gemaakt bij de gerenommeerde search engines⁴ op het Web. Dit alles gaat natuurlijk in mindere mate op voor een website op het intranet. Het gebruik van de website zal hierbij dienen te worden gecommuniceerd en gestimuleerd via interne organisatiemediën.

Snelheid

Beschikbaarheid is nauw verbonden met snelheid. In de praktijk blijkt veelvuldig dat het nog niet goed mogelijk is informatie volledig multimediaal te presenteren vanwege de technische tekortkomingen van het net. Factoren die hier een rol spelen zijn overbelasting en beperkte bandbreedte van de datacommunicatie. Gebruikers verwachten desondanks dat web-media snel benaderbaar zijn en dat ze vervolgens snel van pagina naar pagina kunnen navigeren. Te veel of te grote grafische elementen (afbeeldingen) op een pagina kunnen het oproepen ervan vertragen. Dit is een reden voor gebruikers om af te haken. Het gebruik van data-intensieve informatievormen als graphics, geluid, beeld en video dient daardoor afgewogen te worden met de doorgaans voor gebruikers beschikbare bandbreedte.

In tabel 1 zijn eerder beschreven succesfactoren opgesomd.

Meerwaarde op organisatieniveau

Op organisatieniveau speelt de vraag of een website kan leiden tot een betere ondersteuning van bedrijfsprocessen. Bedrijfsprocessen kunnen worden onderverdeeld in primaire en ondersteunende processen, zoals die worden onderscheiden in de waardeketenmethodologie ([Port85], figuur 4).

De waardeketen toont de totale waarde en bestaat uit waardeactiviteiten en marge. Waardeactiviteiten zijn de op fysiek en technologisch gebied te onderscheiden activiteiten die een organisatie uitvoert. Het zijn de bouwstenen waarmee een organisatie producten en diensten creëert die waardevol zijn voor haar afnemers. Het primaire proces van een organisatie is het productieproces dat haar bestaansrecht geeft. De hierboven liggende balken representeren de ondersteunende bedrijfsprocessen die de

benodigde bedrijfsmiddelen beschikbaar stellen en beheren voor het succesvol functioneren van de organisatie. Deze activiteiten ondersteunen elkaar en de primaire activiteiten door te zorgen voor ingekochte inputs, technologie, menselijk kapitaal en diverse andere bedrijfsomvattende functies.

Om de meerwaarde van web-media theoretisch te onderbouwen en beter in kaart te brengen, kan gebruik worden gemaakt van de bovenbeschreven waardeketenmethodologie. De meerwaarde van een website zal dan tot uitdrukking moeten komen in vergroting van de marge in de waardeketen. Vergroting van de marge betekent vergroting van de waardeproducten en/of verbeteringen/kostenbesparingen in het productieproces.

Communicatietoepassingen

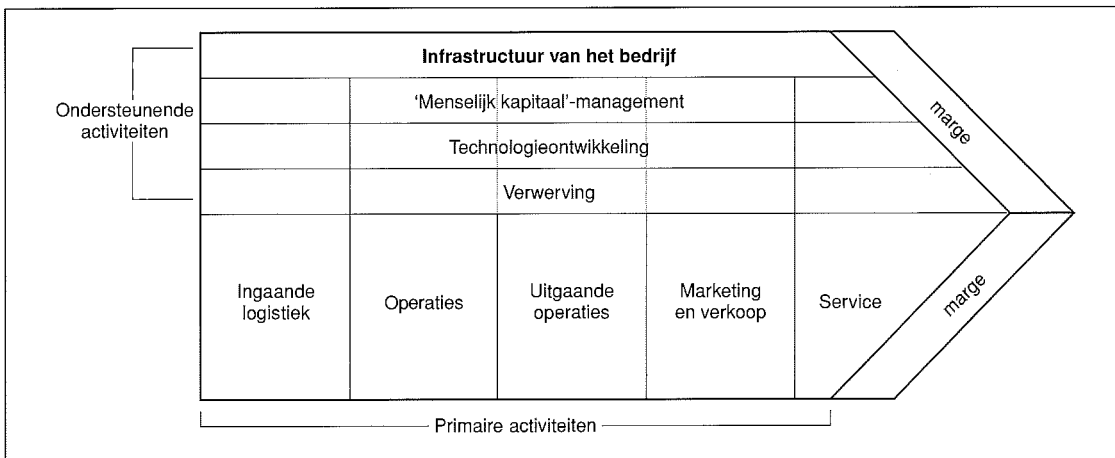
Web-media zijn multimediatoepassingen. Hoewel op organisatieniveau multimediatoepassingen over drie succesfactoren beschikken, te weten communicatie, training en integratie ([Bure95]), is de succesfactor van web-media op het Internet vooral communicatie. Organisaties bevinden zich in een stadium dat ze de communicatie met afnemers op een andere wijze laten verlopen ([Scho96]). In een omgeving waarin informatie en communicatie steeds belangrijker worden, bieden websites als communicatiemediën nieuwe mogelijkheden om de marge in de waardeketen te vergroten. De meerwaarde van websites is bijgevolg te vinden in communicatietoepassingen, zoals geprojecteerd op de primaire processen in de waardeketen van figuur 5.

Voor het verkopen van producten, diensten en informatie en het verlenen van informatieve service wordt het Internet een steeds geschikter medium. Naast de praktische voordelen van elektronische media boven gedrukte media als snelheid, flexibiliteit en kostenreductie, kan een website de effectiviteit van de communicatie verhogen. Door het bieden van interactiviteitsmogelijkheden in een website kan de betrokkenheid van een gebruiker bij de communicatie toenemen, wat erin kan resulteren dat bij promotionele communicatie de boodschap langer wordt onthouden dan bij andere vormen van informatieverspreiding. Het Internet biedt daarbij een bereikbaarheid van vierentwintig uur per dag.

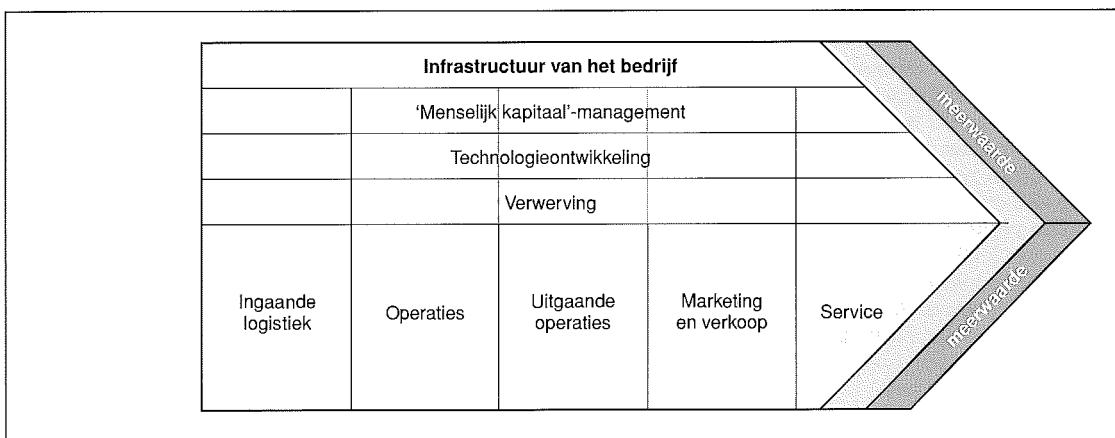
⁴ Zoals Altavista, Yahoo, InfoSeek, HotBot, Excite, Lycos en Ilse.

Aantrekkelijkheid	Interactiviteit	Beschikbaarheid
Vormgeving <ul style="list-style-type: none"> - helderheid/duidelijkheid - gebruikersvriendelijkheid - consistentie - voorspelbaarheid - herkenbaarheid - (grafische) continuïteit - eenduidigheid - gematigdheid in toevoegingen Inhoud <ul style="list-style-type: none"> - betrouwbaarheid - attractiviteit - bruikbaarheid/zinnigheid - doordachttheid leidmotief - actualiteit - eigenaarschap 	Toegankelijkheid <ul style="list-style-type: none"> - evenwichtigheid zoekstructuur - effectiviteit zoekfaciliteiten Betrokkenheid <ul style="list-style-type: none"> - effectiviteit responsystemen - personificatie - adaptiviteit 	Bereikbaarheid <ul style="list-style-type: none"> - logische DNS-naamgeving - aanwezigheid op referentiesites - aanwezigheid op search engines Snelheid <ul style="list-style-type: none"> - duur van responstijd - snelheid informatieoverdracht

Tabel 1. Auditaspecten van een website.



Figuur 4.
De organisatie als waardeketen.



Figuur 5.
Meerwaarde web-media op organisatieniveau.

Steeds meer organisaties met een complexe interne informatiestroom kiezen vanwege de relatieve eenvoud en bovenal vanwege de tijdigheid voor een intranet-website als medium voor de interne communicatie. De uniforme manier waarop informatie via een website op het intranet onafhankelijk van de gebruikte technische platformen aan de interne gebruikersgroep kan worden getoond, is zeer geschikt om de onderlinge communicatie in een organisatie te verbeteren. Een intranet is relatief eenvoudig te realiseren. Omdat het op een bestaande LAN-omgeving kan worden opgebouwd, zijn de initiële kosten bovendien laag. Deze voordelen maken het aantrekkelijk een intranet een stuk van de bestaande informatiestromen rond en binnen een organisatie te laten overnemen. Algemeen geldt voor een intranet dat tijdige beschikbaarheid belangrijker is dan aantrekkelijkheid. Een intranet in een LAN-omgeving heeft voorlopig nog een aanzienlijk hogere snelheid dan het publieke Internet. Dit biedt de mogelijkheid om meer gebruik te maken van geavanceerde toepassingen die een hogere bandbreedte van de communicatielijnen eisen.

Naast het verbeteren van de communicatie kan een andere doelstelling van een website imagebuilding zijn, als onderdeel van de public-relationstrategie. Maar ook dan zal de inzet van het Web als communicatiekanaal een weloverwogen keuze dienen te zijn. Gebruikers maken zich vaak op basis van de website een voorstelling van de organisatie. *Het succes van het medium hangt daarbij af van de mate waarin de succesfactoren aantrekkelijkheid, interactiviteit en be-*

schikbaarheid worden benut. Het is belangrijk hieraan een goede invulling te geven.

Behoud van meerwaarde

Om meerwaarde te behouden is het van belang dat na implementatie de website en de geboden informatie worden onderhouden. Het is vooral het onderhoud dat de kosten van een website bepaalt. De website dient actueel te worden gehouden door regelmatige verversing van de aangeboden informatie. Het toenemend aanbieden van interactiviteitsvoorzieningen doet de onderhoudskosten daarbij evenredig stijgen. Daarnaast dient een website continu beschikbaar en technisch stabiel te functioneren. Wanneer bepaalde onderdelen van een website niet functioneren, zal de kwaliteit van de gehele website sterk negatief worden beïnvloed, waardoor gebruikers zullen afhaken.

Mogelijkheden en beperkingen van web-media zullen door de voortschrijdende technologische ontwikkelingen continu dienen te worden geïnventariseerd en worden afgewogen tegen de wensen van de gebruikersgroep ten aanzien van de mogelijkheden van web-media. Naast een voortdurende analyse van de externe omgeving van de organisatie zal bij elke verandering van de inhoud van de website moeten worden bekeken wat de impact is op de hoeveelheid middelen die moet worden vrijgemaakt om de communicatie optimaal te laten verlopen. De moeilijkheid bij een dergelijke kosten-batenanalyse is dat de baten lastig in geld zijn uit te drukken en vaak pas op langere termijn optreden.

Ontwikkelmethodiek

Web-media kenmerken zich veelal door een voortdurende veranderlijkheid van vormgeving, interactiviteits- en beschikbaarheidsvoorzieningen. In hoog tempo wisselende behoeften van gebruikers en toenemende mogelijkheden van presentatietechnologieën maken het noodzakelijk de inrichting van de website snel te kunnen wijzigen. De hiervoor te hanteren ontwikkelmethode dient hiermee rekening te houden.

Herontwerp van een website vereist een verkort ontwikkeltraject.

Hoewel bij het initieel opzetten van een website traditionele systeemontwikkelingsmethoden overeenkomstig de watervalmethodiek kunnen worden gehanteerd, is het aanbevelenswaardig voor het herontwerpen van de website een evolutionaire ontwikkelmethodiek te hanteren. Het continue evolutieproces van websites en het daarbij toenemende belang van creatief onderscheidend vermogen vereisen een verkort ontwikkeltraject waarbij gebruikerswensen leidend dienen te zijn. De Rapid Application Development (RAD)-methodiek komt aan deze eisen tegemoet door het iteratieve en evolutionaire georiënteerde ontwikkelproces, met kortere doorlooptijd, waarin (vertegenwoordigde) gebruikers volop participeren.

Kritische opmerkingen

Ondanks de commerciële mogelijkheden van web-media zijn gebruikers vooralsnog terughoudend bij de aanschaf van goederen en diensten via het Internet. Dit vanwege de beperkte betaalmethoden en het gebrek aan vertrouwen in de beveiliging daarvan. Veilige betaalmethoden zijn nog in ontwikkeling. Daarbij heerst eveneens terughoudendheid waar het gaat om het verstrekken van persoonlijke gegevens aan een website, waardoor een interactieve dialoog en het leveren van maatwerk wordt bemoeilijkt. Een gedragsverandering dient nog plaats te vinden.

Web-technologie

Als gevolg van het toenemende gebruik van het Web voor informatieoverdracht zijn ook technologieën voor de opslag, verwerking en distributie van de hiervoor benodigde gegevens geëvolueerd. Deze paragraaf zal hierop nader ingaan.

Client/server-technologie

Het Web is conform vele conventionele databasesystemen opgebouwd volgens een client/server-architectuur. Web-media onderscheiden zich echter van conventionele client/server-databases door het gebruik van andere communicatie- en presentatietechnologieën. Web-media maken gebruik van op het Internet gebaseerde technologieën als TCP/IP en HTTP voor de informatieoverdracht tussen client en server. De opmaak en inhoud van de informatieoverdracht zijn opgebouwd volgens het HTML-formaat. HTML-pagina's fungeren als eenheid voor de informatieoverdracht. Voor elke HTML-pagina (en

afbeelding daarin) wordt een nieuwe verbinding tussen de client en de server opgezet en vervolgens weer verbroken.

De verbinding tussen een web-client en een webserver vindt plaats aan de hand van hiërarchische bewegwijzering zoals die door het Domain Name System (DNS) wordt aangeleverd. Als een client verbinding zoekt met een server die niet tot het eigen domein behoort, zal de lokale name-server het adres opvragen bij andere name-servers. Omgekeerd zal de server het DNS raadplegen voor het achterhalen van een client-naam.

Web-browsers

Web-clients dienen te beschikken over een webbrowser om HTML-pagina's weer te geven. Webbrowsers interpreteren een HTML-pagina en laten de gebruiker deze pagina zien in de HTML beschreven opmaak. De presentatiestijl is hierbij browserafhankelijk. Het grote voordeel van een webbrowser is dat deze fungeert als een verzamelbak voor voorheen verschillende informatietoepassingen. Waar eerst afzonderlijke applicaties als spreadsheets, tekstverwerkers en databases moesten worden gebruikt, kan nu de informatieoverdracht in één webbrowser worden gepresenteerd, al dan niet gebruikmakend van zogeheten plug-ins. De voornaamste web-browsers zijn op dit moment Netscape Navigator en Microsoft Internet Explorer.

Webservers

Webservers – de onderliggende computersystemen van een website – fungeren als communicatiepoorten tussen web-browsers en gegevensbestanden. Hierbij vindt informatieoverdracht plaats aan de hand van HTML-pagina's. De HTML-pagina's zijn daarbij benaderbaar aan de hand van een Uniform Resource Locator (URL). Het gebruik van URL's is een standaard om de locatie van web-pagina's op het Web aan te geven. De URL omvat het locatieadres van de webserver, aangevuld met een directory-aanduiding waarop de desbetreffende HTML-pagina is gelokaliseerd (zoals <http://www.bedrijfsnaam.nl/product/demo.htm>).

Statische en dynamische informatieoverdracht

Web-informatie kan bestaan uit statische of dynamische HTML-pagina's. Het merendeel van de informatie op het Web is momenteel statische informatie. Bij statische pagina's wordt informatie door de webserver volgens een vast stramien in een HTML-pagina weergegeven. Statische informatieoverdracht wordt toegepast als de informatie niet vaak verandert en voor de inhoud van de informatiestroom geen interactie met de gebruiker noodzakelijk is.

HTML kan naast informatievormen ook objecten als executeerbare scripts bevatten. Een script is een klein softwareobject, dat in staat is de web-browser en de webserver te instrueren bepaalde handelingen uit te voeren. Een script maakt het mogelijk aan HTML-pagina's dynamisch gedrag toe te voegen, wat wil zeggen dat de inhoud van een gepresenteerde pagina afhankelijk kan worden gemaakt van gebruikershandelingen. Zo kan een script de web-browser geformatteerde schermen met invoervelden laten tonen en na invoer de gegevens naar de webserver laten sturen, die de data vervolgens doorsluisst naar

een databasesysteem. Evenzo kan het databasesysteem HTML-pagina's wijzigen, om actuele gegevens en reacties op gebruikersacties te tonen. Scripts breiden het statische HTML-concept zodoende uit met dynamische gegevensinteractie, waardoor afhankelijk van de gebruikersinteractie gegevens kunnen worden getoond en ingevoerd.

Dynamische pagina's zijn absoluut noodzakelijk voor serieuze website-toepassingen. Ze kunnen twee dingen die met statische pagina's onmogelijk zijn. Ten eerste kan de gebruiker een database doorzoeken volgens individuele criteria, waarbij een pagina met variabele inhoud wordt gemaakt op het moment dat het nodig is. Ten tweede zijn dynamische pagina's geschikt voor het invoeren van gegevens door de gebruiker. Voor het realiseren van dynamische informatieoverdracht dient behalve van een web-browser en webserver evenwel gebruik te worden gemaakt van additionele technologieën. Deze technologieën worden navolgend besproken.

Gateways

Voor het genereren van dynamische pagina's dient een webserver te beschikken over programmeerfaciliteiten. De standaard-interfaceomgeving is hiervoor de Common Gateway Interface (CGI). CGI beschrijft de technische afspraken om databasesystemen te verbinden met webserver. Webserver kunnen op de achtergrond programma's draaien en vervolgens de resultaten daarvan naar de gebruiker toezenden. De server gebruikt hiervoor een CGI-script om de binnengekomen informatie te verwerken en de output - vertaald in HTML-pagina's - te communiceren naar de web-browser. Hiertoe beschrijft een CGI-script hoe de invoer en uitvoer moet worden geregeld. Zo kunnen CGI-scripts bijvoorbeeld queries uitvoeren voor een database, e-mail versturen, overzichten maken, transacties uitvoeren en andere programma's opstarten. Uitvoering van een CGI-script vindt daarbij plaats op de server. CGI-scripts kunnen in nagenoeg iedere programmeertaal worden geschreven⁵.

Middleware

Een gateway wordt vaak aangevuld met zogeheten middleware-tools⁶. Middleware is de communicatiepoort tussen de gateway-interface en het databasesysteem. Verzoeken van CGI-scripts tot het benaderen van een databasesysteem worden door middleware vertaald in (SQL-)queries (en/of stored procedures) en gecommuniceerd naar het relevante databasemanagementsysteem (DBMS). Wanneer

een query door het DBMS is uitgevoerd, worden de outputgegevens door de middleware gecommuniceerd aan de CGI-scripts. Het CGI-script verwerkt de outputgegevens in een HTML-pagina⁷ - vergelijkbaar met mail-merge in een tekstverwerker - en communiceert het resultaat naar de web-browser.

In figuur 6 is de voor dynamische informatieoverdracht benodigde architectuur weergegeven.

Beperkingen CGI

De beschreven architectuur met gebruikmaking van CGI is een vertrekpunt om web-media te ontwikkelen. Het voordeel van CGI is dat het als standaard door alle webserver wordt ondersteund. Een nadeel van een common interface is dat het in feite buiten de webserver om - in een afzonderlijke adresruimte - wordt geïmplementeerd, waardoor een zekere prestatieoverhead ontstaat. Hierdoor hebben web-media gebaseerd op CGI-scripts een verminderde performance. Om de vertaalslag te maken tussen databasegegevens en HTML-pagina's hebben leveranciers van serversoftware vaak een eigen snellere interface met geïntegreerde middleware ontwikkeld. In tabel 2 is hiervan een overzicht opgenomen.

De gatewayprogrammatuur is operationeel op de server. Dit betekent dat de systeembelasting ook volledig op de server plaatsvindt. De web-browser fungeert hierbij slechts als doorgeefluik tussen de gebruiker en de webserver. Voor iedere door de web-browser gepresenteerde pagina - of objecten daarin - is een sessie met de server noodzakelijk. Een eenvoudig invulformulier zal hierdoor goed te realiseren zijn, maar voor serieuze web-media met een hoge mate van interactieve informatieoverdracht zal de functionaliteit van de web-browser beter dienen te worden benut door gebruikmaking van zogeheten scripts en applets.

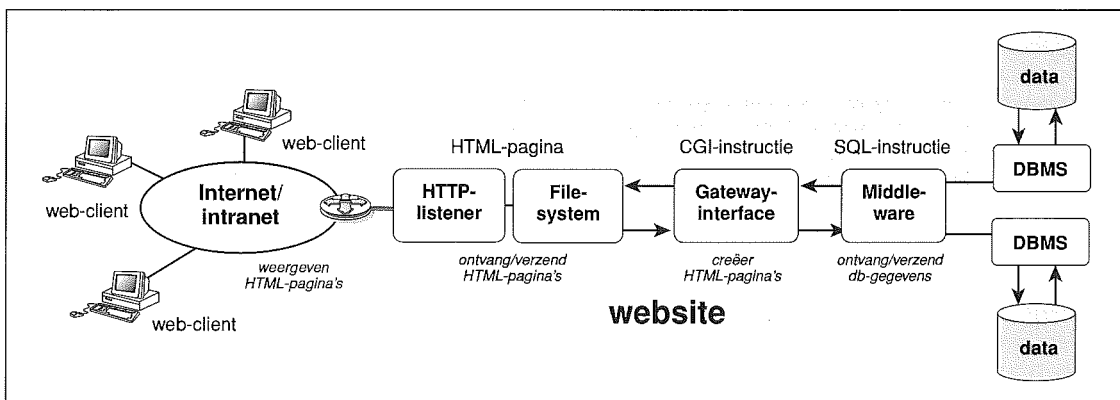
Scripts en applets

Naast CGI-scripts die op de server operationeel zijn, kunnen met HTML-pagina's ook scripts worden meegestuurd die operationeel worden op de client web-browser. Hierdoor kan meer dynamiek aan HTML-pagina's worden toegevoegd zonder aparte CGI-scripts te ontwikkelen en wordt de server ontlast. Pagina's kunnen daarbij naast eenvoudige standaard-HTML-scripts voor bijvoorbeeld gegevens- en invoervalidatie, ook geavanceerdere objecten bevatten. Waar HTML eventueel tekortschiet inzake gebruikersinteractie, voorzien verschillende leveranciers van Internet-technologie nu

5 De hiervoor meest gebruikte programmeertalen zijn C, Perl en Visual Basic.

6 Open DataBase Connectivity (ODBC) en Java DataBase Connectivity (JDBC) zijn hieraan bekende voorbeelden.

7 HTML-pagina's gevuld met databasegegevens worden HyperText Extension (HTX-) files genoemd.



Figuur 6. Architectuur van een website met dynamische informatieoverdracht.

Tabel 2. Marktoverzicht gatewayprogrammatuur.

Product	Leverancier	Nadere informatie
Active Web	Active Software	www.activesw.com
Amazon	Intelligent Environments	www.ieinc.com
Cold Fusion	Allaire Corp.	www.allaire.com
DB2 WWW Connection	IBM	www.software.ibm.com
dbWeb	Microsoft	www.microsoft.com
Domino	Lotus	domino.lotus.com
HAHTsite	HAHT Software	www.haht.com
IDC/HTX Connector	Microsoft	www.microsoft.com
IDS Server	IDS Software	www.idssoftware.com
Internet Information Server	Microsoft	www.microsoft.com
IntraBuilder	Borland	www.borland.com
Intranet Database Assistant	Intranet 2001	www.inet2001.com
Jdesigner Pro	BulletProof	www.bulletproof.com
LiveWire Pro	Netscape	home.netscape.com
NetData	IBM	www.software.ibm.com
NetDynamics	NetDynamics	www.netdynamics.com
NetObjects	NetObjects	www.netobjects.com
Oracle WebServer	Oracle	www.oracle.com
Quest	Level 5	www.l5r.com
Sapphire/Web	Bluestone	www.bluestone.com
Tango Enterprise	EveryWare	www.everyware.com
VirtuFlex	VirtuFlex	www.virtuflex.com
Visual InterDev	Microsoft	www.microsoft.com
Web+	TalentSoft	www.talentsoft.com
Web Builder	VPE Inc.	www.vpe.com
Universal Web Connect	Informix	www.informix.com
WebDBC	StormCloud Development	www.stormcloud.com
WebMate/Foundation	WebMateTechnologies	www.webmate.com
WebObjects	NeXT Software	www.next.com
WebSpeed	Progress	www.progress.com
WebSQL	Sybase	www.sybase.com

⁸ Zoals bijvoorbeeld Java of Visual Basic.

⁹ De web-browser van Microsoft ondersteunt Java, maar Microsoft heeft ook een alternatief op de markt gezet: ActiveX. ActiveX maakt gebruik van programmeertalen als Visual Basic en Visual Java. Een daarmee ontwikkeld programma wordt control genoemd. ActiveX controls zijn objecten die worden ingevoegd op een pagina. ActiveX controls zijn specifiek voor Windows geschreven en maken daardoor dan ook goed gebruik van die architectuur. Dit vertaalt zich in meer flexibiliteit en performance dan universele Java-applets. In tegenstelling tot Java-applets zijn ActiveX controls alleen uitvoerbaar op een Windows-platform en hierdoor niet platform-onafhankelijk.

in aanvullende programmeertalen⁸ voor het creëren van deze objecten.

Hierbij wordt onderscheid gemaakt tussen scripts en applets. Een script is een verzameling instructies waarbij de broncode in de HTML-pagina is opgenomen. Een script wordt bij uitvoering geïnterpreteerd door een web-browser. Een applet is verzameling instructies die in gecompileerde vorm met een HTML-pagina wordt meegestuurd en wordt uitgevoerd door de web-browser. Het grote verschil tussen scripts en applets enerzijds en CGI-scripts anderzijds is dat de werking op de client plaatsvindt en niet op de server. Een script of applet kost, als deze eenmaal is opgehaald door de web-browser, geen netwerkbandbreedte en verwerkingstijd op de server meer. Dit biedt allerlei nieuwe geavanceerde toepassingsmogelijkheden op gebieden als interactie en animatie.

Platformonafhankelijkheid

Applets worden hoofdzakelijk in Java ontwikkeld. Java is een op C++ geënte programmeertaal die de laatste tijd zeer in de schijnwerpers staat. Deze taal

onderscheidt zich van andere programmeertalen doordat de uitvoering van de programma's onafhankelijk is van het onderliggende platform. Een Java-applet wordt niet vertaald in machinecode, zoals normaliter bij applicaties, maar wordt vertaald naar een platformonafhankelijk bytecode-formaat. Om deze bytecode uit te kunnen voeren dient de web-browser te beschikken over een ingebouwde Java-interpretter, een zogeheten Java Virtual Machine (JVM). De JVM vormt de vertaalslag naar het onderliggende besturingssysteem. Web-browsers⁹ beschikken tegenwoordig over een ingebouwde JVM, waardoor de uitvoering van een Java-applet onafhankelijk is van zowel het besturingssysteem als het hardwareplatform (althans alle platformen met besturingssystemen met een JVM).

TOEKOMSTIGE ONTWIKKELINGEN

Het is moeilijk te voorspellen hoe web-media in de toekomst gaan worden ingezet en gebruikt. Wel is

waarschijnlijk dat in de succesfactoren enige veranderingen zullen optreden.

De aantrekkelijkheid van media zal langzamerhand vanuit de perceptie van de gebruiker steeds meer afnemen. Er treedt een zogenaamd gewenningsproces op. Een kwalitatief uitgewerkte multimediale aantrekkelijkheid zal steeds meer een basisvoorwaarde voor een website gaan worden. De mate van interactiviteit zal daarbij toenemen. Web-media zullen meer gaan anticiperen op de keuzen van de gebruiker en zich steeds meer aan het denkproces van de gebruiker aanpassen. Een hiervoor benodigde gedragsverandering zal voorafgaand overigens nog moeten plaatsvinden.

Hierbij zullen standaard-inrichtingsprincipes voor web-pagina's tot ontwikkeling komen. Web-pagina's en hyperlinks maken een evolutie door vergelijkbaar met die van het gedrukte boek. Onze huidige opvattingen met betrekking tot het structureren van informatie in elektronische vorm stammen nog grotendeels af van de inrichtingsprincipes die gebruikt worden voor het maken van boeken. Elk onderdeel van een boek, van de inhoudsopgave tot aan de index en de voetnoten, is in de loop der eeuwen geëvolueerd ([HTML97]). Na de publicatie van de Gutenberg-bijbel in 1456 duurde het meer dan honderd jaar voordat zaken als een titelpagina of paginanummering gemeengoed werden.

Veel ontwikkelingen zullen plaatsvinden ter verbetering van de beschikbaarheid van web-media. Vooral het verhogen van de bandbreedte voor snellere informatieoverdracht over het publieke Web heeft hoge prioriteit¹⁰. Web-technologieën zullen zich daarbij verder evolueren. Traditionele client/server-technologieën voor informatieoverdracht zullen als gevolg hiervan in toenemende mate worden overvleugeld door web-technologie. Intranetten zullen de ruggengraat vormen van toekomstige bedrijfsautomatisering. Geografisch gespreide organisaties zullen hierbij een intranet simuleren door het Internet te gebruiken voor het opzetten van een zogeheten Virtual Private Network¹¹. Voor de consumentenmarkt zal de beschikbaarheid kunnen toenemen door inschakeling van het huidige breedbandige kabelnet. Daarbij wordt zelfs gedacht aan het volledig integreren van de web-omgeving met het televisietoestel, wat bekend staat onder Web-tv¹².

Daarnaast zal steeds meer aandacht worden besteed aan de bereikbaarheid en de toegankelijkheid van het Web. Het aantal nieuwe sites op het publieke Web groeit gestaag. Omdat het daarbij eenvoudig en verleidelijk is om web-media uit te breiden groeit de omvang van de op het publieke Web beschikbare pagina's explosief. Het aantal pagina's op het Web wordt nu al geschat op om en nabij de 150 miljoen. Het huidige URL-systeem waarbij een web-document is gelokaliseerd op basis van de server- en documentnaam, zal daarbij – door de beperkte onderhoudsmogelijkheden – waarschijnlijk op middellange termijn worden vervangen door een systeem dat uitsluitend werkt met een documentnaam, onafhankelijk van de fysieke serverlocatie. Door de groei in aantal web-pagina's zal evenwel het actueel houden en het toevoegen van nieuwe informatie een taak van formaat worden. Geavanceerde beheertools hiervoor zijn in ontwikkeling.

Als gevolg van de groei in aantal web-pagina's krijgen gebruikers tijdens het navigeren over het Web in toenemende mate een informatieoverload te verwerken. Zogeheten push-technologie zal hierin verandering kunnen brengen. Gebruikers gaan vandaag de dag zelf op zoek en trachten – al dan niet met behulp van search engines – de benodigde informatie aan het Web te onttrekken. Het Web fungeert zodoende als een zogeheten 'pull'-medium. In de nabije toekomst zal het Web zich meer ontwikkelen van een 'pull'- naar een 'push'-medium. Dat wil zeggen van het passief opvragen van informatie door gebruikers naar het versturen van informatie door websites aan geselecteerde gebruikers binnen een bepaalde doelgroep. Dit wordt ook wel web-casting¹³ genoemd. Push-technologie is een nieuwe benadering van informatieoverdracht en zal een verandering teweegbrengen in de perceptie en het gedrag van gebruikers in de omgang met informatie op het Web.

In de nabije toekomst zal het Web zich ontwikkelen van 'pull'- naar 'push'-medium.

De huidige gegevensstromen op het Web bestaan nog vooral uit het eenvoudig overdragen of aandragen van (dynamische) informatie. In de toekomst zullen in toenemende mate naast informatie ook bijbehorende softwareobjecten (applets) worden aangeleverd, wat installatie van applicaties op de client overbodig maakt. Java gaat hierbij als programmeertaal voor platformonafhankelijke applets ongetwijfeld een belangrijke rol spelen. De object-georiënteerde principes van deze programmeertaal beloven daarbij een sterke verkorting van trajecten voor de ontwikkeling van nieuwe software. Technologieën voor het volledig realiseren van een platformonafhankelijke gedistribueerde objectomgeving¹⁴ zijn momenteel in ontwikkeling.

Als dan toch alle applicatieve programmatuur met de gegevens kan worden meegestuurd, is het voor een werkstation eigenlijk alleen nog maar noodzakelijk te beschikken over een web-browser met een Java-interpret (JVM) daarin. Deze gedachte staat bekend als het Netwerk Computer (NC)-concept¹⁵. NC's zijn eenvoudige werkstations, enkel voorzien van browser-programmatuur met ingebouwde JVM. Breed uitgemeten voordelen van de NC ten opzichte van de traditionele personal computer zijn kostenbesparingen in de aanschafprijs en sterk verminderde onderhouds- en beheerinspanningen, doordat het onderhoud van de automatisering binnen organisaties (weer) wordt gecentraliseerd. Daarbij wordt gescandeerd dat de huidige PC in zakelijke omgevingen het daarbij binnen vijf jaar zou afleggen tegen NC. Het NC-concept gaat echter in wezen weer terug naar het model van de 'dikke client', met veel code in de Java-applets. Aangezien dit model mede verantwoordelijk is geweest voor het hoge aantal mislukte client/server-projecten, zullen de voordelen van het NC-model derhalve nog ondubbelzinnig moeten worden aangetoond.

Web-browsers zullen op den duur de grootste ge-

10 Momenteel wordt geëxperimenteerd met het zogeheten 'Internet2'. Door gebruikmaking van een nieuw IP-protocol (IPv6) kunnen daarbij snelheden van 2,4 gigabyte per seconde worden bereikt.

11 Termen die in dit verband ook worden gebruikt, zijn 'Extranet', 'gekoppelde intranets' en bij een koppeling met leveranciers/klanten: 'open EDI'.

12 Of de Web-tv een succes wordt is vooralsnog onduidelijk. Verschillende markt-onderzoeken spreken elkaar duidelijk tegen. Desondanks vinden momenteel door leveranciers als Philips, Sony en Microsoft omvangrijke investeringen plaats ter ontwikkeling van de Web-tv.

13 Bedrijven als Pointcast hebben hiermee inmiddels naam gemaakt. Sinds kort passen ook Netscape en Microsoft push-technologie toe in hun web-browsers.

14 Bekend als de CORBA-architectuur, waarin is gedefinieerd hoe de onafhankelijke objecten met elkaar dienen te communiceren.

15 Andere benamingen zijn Net-PC, domme PC, Java-station en Windows-Based Terminal.

Drs. A.M. Buren

Is afgestudeerd in de doctoraalopleidingen Bestuurlijke Informatiekunde en Bedrijfs-economie en is momenteel werkzaam bij KPMG EDP Auditors te Amstelveen. Hij heeft zich in de afgelopen jaren verdiept in ontwikkelingen op het gebied van informatieoverdracht en multimedia. Zijn aandachtsgebieden liggen bij ondernemingsbrede beveiligingsvraagstukken, advies voor en audit van database- en web-technologieën. Thans volgt hij de postdoctorale EDP-Audit opleiding aan de Vrije Universiteit Amsterdam.

mene deler vormen tussen PC's en NC's. De verdere ontwikkeling hiervan zal grotendeels in handen zijn van de softwareontwikkelaars Netscape en Microsoft. Het is rumoerig in deze markt. Omdat deze twee de browsermarkt feitelijk in handen hebben, zullen ze daarbij al te graag eigen extensies aan HTML toevoegen. Eigenlijk doen ze dit al, net als databaseleveranciers met SQL hebben gedaan.

Het Web beschikt niettemin over belangrijke standaardisatieorganen zoals het Internet Engineering Task Force (IETF) en het World Wide Web Consortium (W3C) om deze pogingen te pareren. De consortia zullen de verdere ontwikkeling van het Web gestalte gaan geven door het HTML uit te breiden en het Web geschikter te maken voor commerciële transacties. HTML is slechts het begin en zal verder moeten doorgroeien. De huidige HTML-standaard heeft een aantal beperkingen. Zo is HTML primair tekstgeoriënteerd, is de opmaak onlosmakelijk met de weergave verbonden en is de presentatiestijl vastgelegd in de functionaliteit van de browser. Daarnaast heeft HTML geen structuur; iedere pagina is eenvoudigweg een verzameling bytes. Het W3C-comitee heeft inmiddels een nieuwe gestructureerde standaard gedefinieerd met de eXtended Markup Language (XML), die tegemoetkomt aan deze beperkingen.

Afsluitend

De mogelijkheden die het Web biedt, zijn voor veel bedrijven zo langzamerhand wel duidelijk. Het Internet vormt een nieuw communicatiekanaal voor aangeboden producten, diensten en informatie. Het intranet biedt daarbij mogelijkheden tot effectievere interne berichtgeving. Mogelijkheden die – door rekening te houden met een kwalitatieve opzet – meerwaarde kunnen bieden ten opzichte van andere vormen van informatieverbreiding.

Tegenover de kansen van web-media staan momenteel evenwel nog omvangrijke veiligheidsrisico's. Risico's zoals het onderscheppen van vertrouwelijke gegevens door derden, ongeoorloofde toegang tot bedrijfssystemen, virusaanvallen en ongeoorloofde wijziging van web-pagina's en onderliggende databasegegevens zijn op het Web duidelijk aanwezig.

Beveiliging wordt de grootste opgave. Het Internet moet echt veilig zijn, wil het als universeel communicatiemedium worden gebruikt voor electronic commerce-doeleinden. Hoewel in toenemende mate veiligheidsmaatregelen¹⁶ in de huidige web-tech-

nologieën worden geïntegreerd, blijven de gebruikers daarbij de cruciale factor. Gebruikers moeten de veiligheid van het Internet ervaren. De nabije toekomst zal moeten uitwijzen of dit gevoel van veiligheid kan worden gerealiseerd.

Ondanks de veiligheidsrisico's besluiten steeds meer bedrijven het Web als communicatiemedium te gebruiken. Investeringslijnen daarbij bestaand te zijn tegen toekomstige ontwikkelingen doordat inmiddels vrijwel alle neuzen dezelfde kant op staan. Het gebruiksgemak ligt op hoog niveau en Internet-technologie is compatibel te maken met de bestaande techniek.

Ondanks de huidige beperkte beveiligingsmogelijkheden en bandbreedte, en de niet te onderschatten informatieoverload, zullen web-toepassingen toeneemen en daarbij veelzijdiger worden. De behoefte aan, of in ieder geval de belangstelling voor het Web is momenteel groot. Als gevolg van die behoefte zal het Web alleen nog maar groeien, in omvang en in mogelijkheden.

LITERATUUR

[Bure95] A.M. Buren, *Multimedia nader bekeken*, Compact 1995/2.

[Dort96] A. Dortland, *Nieuwe media en ontwikkelingen in de technologie*, KPMG Samsom, Alphen aan den Rijn 1996.

[HTML97] *HTML, the complete guide*, <http://www.magnetic.demon.co.uk/>, 1997.

[Paan96] P. Paans, *Internet en beveiliging*, in: *Internet voor accountants*, VERA-seminar, 3 december 1996.

[Port85] M.E. Porter, *Competitive Advantage: creating and sustaining superior performance*, The Free Press, London 1985.

[Scho96] T. Schotgerrits, *De nieuwe media consument, ontwikkelingen op het gebied van nieuwe media en de betekenis daarvan voor organisaties*, KPMG Samsom, Alphen aan den Rijn 1996.

[SUN96] Sun Microsystems, *Guide to web style*, <http://www.sun.com:80/styleguide/>, 1996.

16 Zoals uitbreiding van het huidige HTTP-protocol met cryptografische technieken voor het versleutelen van bijvoorbeeld PIN-codes of creditcardnummers en authenticatiemechanismen om de identiteit van cliënten vast te stellen.

CGI	Common Gateway Interface	NC	Netwerk Computer
DBMS	Database Management Systeem	ODBC	Open DataBase Connectivity
DNS	Domain Name System	OSI	Open Systems Interconnection
FTP	File Transfer Protocol	RAD	Rapid Application Development
HTML	HyperText Markup Language	SQL	Structured Query Language
HTTP	HyperText Transaction Protocol	TCP	Transmission Control Protocol
HTX	HyperText Extension	UDP	User Datagram Protocol
IETF	Internet Engineering Task Force	URL	Uniform Resource Locator
IP	Internet Protocol	W3C	World Wide Web Consortium
JDBC	Java DataBase Connectivity	WWW	World Wide Web
JVM	Java Virtual Machine	XML	eXtended Markup Language

Tabel 3. Afkortingen.



KPMG EDP Auditors
Samsom BedrijfsInformatie