

COMPACT

TIJDSCHRIFT EDP-AUDITING

HET JAAR 2000

1997/1

INHOUDSOPGAVE

Compact ©
Jaargang 24, nummer 1
Een uitgave van KPMG EDP
Auditors NV en Samsom Bedrijfs-
Informatie, werkzaamheidschap van
Wolters Kluwer NV.
Het blad verschijnt 6 x per jaar.

Redactie
Prof. A.W. Neisingh RE RA
(hoofdredacteur)
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.drs. J. van der Vlugt

Adviesraad
Prof.dr. J.C. Arnbak
Mr. P. van Dijken
G. van Essen RA
Prof.mr. H. Franken
Dr. K.J. Mollema RA
Prof. H.B. Moonen RE RA
Prof.dr.ir. R. Paans RE
Redactiesecretariaat
Mv. I. de Koning,
Samsom BedrijfsInformatie,
Postbus 4,
2400 MA Alphen aan den Rijn
Tel.: 0172 - 466 746
Fax : 0172 - 466 569

Vormgeving
Bureau Karnker, Delft
Opmaak
Sander Pinkse Boekproductie,
Amsterdam
Abonnementen
f 165,- per jaar incl. BTW. Losse
nummers f 45,- incl. BTW. Stu-
dentenabonnement 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
Samsom 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



Lid van de Nederlandse organisatie
van tijdschriftuitgevers NOTU

ISSN 0920 - 1645

2

Redactioneel

3

Een overzicht van Y2K-problemen

Ir.drs. J. van der Vlugt

Om binnen de organisatie voldoende aandacht en budget te krijgen voor het oplossen van Jaar 2000-problemen zal een IT-manager of EDP-auditor over voldoende materiedeskundigheid moeten beschikken om de Jaar 2000-projecten te kunnen beheersen en controleren. In dit artikel wordt een overzicht gegeven van optredende problemen, oplossingstactieken en organisatorische aspecten.

18

Testen en de Millennium-conversie

Drs. T. Koomen en drs. E. Broekman CISA

Testen dient een belangrijke activiteit binnen elk Millennium-conversieproject te zijn. Al te vaak wordt echter aan testen niet voldoende continue aandacht gegeven. In dit artikel wordt een gestructureerde aanpak besproken om het testen tot een geïntegreerd onderdeel van ieder Millennium-conversieproject te maken.

29

'2000', een millennium om naar toe te werken

O. Venhuis

Een uniforme aanpak is nodig om de zo omvangrijke en complexe millenniumproblematiek op een beheersbare wijze op te lossen. Dit artikel bespreekt zo'n aanpak, waarbij een risico-analytische inslag voorop staat die leidt tot een systematische aanpak.

39

EDP Auditorium

Prof.mr.dr. J.M. Smits

In de column 'Het werk afgemaakt!' wordt ingegaan op ontwikkelingen rond het intellectueel eigendomsrecht. Pas op met het overnemen van paginacijfers ...

40

Cumulatief

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 advies, 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 redacteuren 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.

Wie denkt geen last te zullen hebben van Jaar 2000-problemen heeft een probleem. Het is niet voor niets dat juist ook onder bedrijven die soms al jaren geleden van start zijn gegaan om de Jaar 2000-problematiek op te lossen een gevoel van haast bespeurbaar is, terwijl wie nog niet met het probleem heeft geworsteld vaak de problemen onderschat.

Kennelijk 'helpt' het de complexiteit en omvang van de problemen van nabij mee te maken om daar een goed beeld van te krijgen. Het is niet voor niets dat de oorspronkelijk weggehoonde schattingen van de omvang van de problemen steeds minder kritiek ondervinden.

Er is geen substituuat voor ervaring. Maar als er één ding duidelijk is dan is het dat de Jaar 2000-problemen op een gestructureerde manier moeten worden aangepakt. Die gestructureerde aanpak kan natuurlijk niet als blauwdruk aan een organisatie worden opgelegd. Wel kunnen specifieke aspecten van de aanpak worden aangedragen om als toetssteen te kunnen dienen.

Eén van die aspecten is testen. Het adagium 'test early, test often' is immers als altijd van kracht. Niet alleen raken de problemen mogelijk, of eerder waarschijnlijk, alle systemen en systeemonderdelen, waardoor ze ten minste alle moeten worden getest. Het is vooral de complexiteit in en tussen de systemen die uitgebreid testen nodig maakt. Het idee dat testen een soort afrondende klus van ondergeschikt belang is, zal dus moeten worden bijgedraaid: testen behoort een geïntegreerd onderdeel te zijn van alle projectfasen.

Het is daarnaast natuurlijk ook zaak om de projecten zodanig in te richten dat ze al bij voorbaat worden gericht op de risico's. Als na een impactanalyse de onderkende risico's op een samenhangende wijze worden aangepakt, kan worden voorkomen dat achteraf, bij het testen, allerlei reparatiewerk nodig blijkt of dat belangrijke risico's niet zijn weggenomen.

Van uitstel kan geen afstel komen. Integendeel, iedereen zal alle zeilen moeten bijzetten om de Jaar 2000-projecten tot een goed einde te brengen. Hopelijk kan deze Compact enige klippen helpen omzeilen.

Ir.dr.s. J. van der Vlucht

Een overzicht van Y2K-problemen

Ir.dr.s. J. van der Vlugt

Zonder inzicht in een aantal technische aspecten is de ernst van de Jaar 2000-problemen moeilijk aan te tonen. Pas als duidelijk is wat waar fout kan lopen, kunnen de juiste oplossingstactieken worden bepaald.

INLEIDING

Tenzij er maatregelen worden genomen nu het nog kan, zult u na de komende millenniumwisseling in het jaar 2000 (Year 2000 ofwel Y2K) van een koude kermis thuiskomen – als u al thuis kunt komen. Mogelijk komt u uw hotel in het zonnige oord waar u de eeuwwisseling viert niet uit omdat de beveiligingssoftware op hol is geslagen. Bovendien: kunt u dan nog met uw creditcard betalen als de betaalapparatuur ontregeld is? Mogelijk werkt de boordcomputer van het vervoer – auto, trein – dat u naar het vliegveld zal brengen niet naar behoren. Hoe zeker bent u ervan dat de apparatuur van uw vliegtuig Jaar 2000-bestendig is?

Bij terugkomst zult u dan te weten moeten komen hoe de organisatie waar u werkt of anderszins uw inkomen van ontvangt, er dan voorstaat. Het is verre van denkbeeldig dat één of meer voor de organisatie kritieke systemen zo veel foutieve output geven dat de bedrijfsvoering in het honderd is gelopen en niet meer aan lopende verplichtingen kan worden voldaan.

Het zal weliswaar niet mogelijk zijn alle maatschappelijke problemen te voorkomen, wel kan ervoor worden gezorgd dat de organisatie zelf al het nodige heeft gedaan. Wat er moet worden gedaan, is nogal eens onduidelijk. In veel gevallen wordt daardoor – mede om politieke, lees: budgettaire, redenen – ten onrechte de omvang van de problematiek onderschat. Om hiertegen te kunnen optreden, zal een IT-manager of EDP-auditor naast de op Internet rijkelijk voorhanden zijnde ‘politieke’ tegenargumenten moeten beschikken over voldoende materiedeskundigheid om de Jaar 2000-projecten te kunnen beheersen en controleren.

Hieronder zal eerst worden uiteengezet welke soorten fouten te onderkennen zijn en waar ze in systemen kunnen optreden. Tevens zullen enkele juridische aspecten worden aangestipt. Vervolgens zullen de hoofdlijnen worden aangegeven van de toepasbare oplossingstactieken.

Daarna zullen de organisatorische aspecten worden toegelicht van het management van de mogelijk zeer omvangrijke, in nauwe samenhang uit te voeren projecten. Van strategiebepaling tot evaluaties verschilt het management van de projecten niet wezenlijk van dat van andere, maar er zijn telkens wel attentiepunten te onderkennen.

Ten slotte wordt een samenvatting gegeven van de argumenten tegen het mogelijk bestaan van een ‘Silver Bullet’-applicatie die alle problemen geautomatiseerd zou kunnen oplossen.

WAAR DE PIJN ZIT

De pijn zit in het gewoontegetrouw gebruiken van slechts twee jaarcijfers in data. Iedereen begrijpt immers dat '97' staat voor 1997. Omwille van efficiëntie van opslag van gegevens zijn de jaartallen ook zo in systemen gebruikt. Nu echter de eeuw-grens zeer nabij is, komt het steeds meer voor dat de voorvoeging '19..' niet meer opgaat. En pas nu blijkt hoe diep die impliciete aanname in de systemen is verankerd.

In deze paragraaf zal een korte uiteenzetting worden gegeven van wat de problematiek is en in welke systeemelementen dat kan gebeuren. Tevens wordt kort ingegaan op enkele juridische aspecten, waaronder aansprakelijkheid.

Optreden fouten

De fouten die op kunnen treden in het (verkeerde) gebruik van (verkeerde) data, worden gemaakt in:

- berekeningen;
- vergelijkingen;
- dataformaten;
- data-opslag;
- speciaal gebruik.

Berekeningen

Het grootste deel van de optredende fouten zal voorkomen in de categorie 'berekeningen'. Rekenfouten zijn op zich niet zo'n groot probleem, ware het niet dat veel van dit soort berekeningen moeilijk als zodanig herkenbaar is. Bovendien kunnen de meest voor de hand liggende oplossingen van geval tot geval verschillen, terwijl omwille van de beheersbaarheid standaardisatie en dus lokaal suboptimale oplossingen nodig zijn. De berekeningen die op data zijn uit te voeren, kunnen worden ingedeeld in de onderstaande categorieën:

- *Het berekenen van de tijdsduur tussen twee data*, bijvoorbeeld rentedagen of een leeftijd van een persoon. In het eerste geval zullen rekenfouten wellicht veel minder opvallen dan in het tweede, maar in beide gevallen zouden uitgebreide handmatige

Tabel 1. Speciale berekeningen voor dataconversies.

| Berekening | Variabelen | Betekenis (CC = eeuwcijfers, YY = jaarcijfers, MM = maandcijfers, DD = dagcijfers) |
|----------------------|-------------------------------|---|
| $A = 10000.01 * B$ | A, B lengte 6, geen decimalen | Indien B = MMDDYY dan wordt A = YYMMDD |
| $A = 100.0001 * B$ | A, B lengte 6, geen decimalen | Indien B = YYMMDD dan wordt A = MMDDYY |
| $A = 10000.0001 * B$ | A, B lengte 8, geen decimalen | Indien B = CCYYMMDD dan wordt A = MMDDCCYY |
| $A = 10000.0001 * B$ | A, B lengte 8, geen decimalen | Indien B = MMDDCCYY dan wordt A = CCYYMMDD |

(Alle indien de hoogste twee cijfers en alle vier decimalen worden getrunceerd.)

controles en correcties geld- en gezichtsverlies moeten beperken. Hoe plezierig is het voor een organisatie als bekend wordt dat zij rente niet goed kan berekenen?

- *Het berekenen van een datum op basis van startdatum en tijdsduur*. Het zal niet in uw dagelijks leven spelen, maar in de Verenigde Staten zijn er staten waar de strafmaat en de uitvoering van straffen door het Jaar 2000-probleem worden beïnvloed: de duur van voorwaardelijkheid van straffen kan niet over 1/1/2000 liggen omdat de systemen dat niet aankunnen. Allicht dat in elke organisatie dergelijke berekeningen vooruit te bedenken zijn.

- *Het berekenen van de dag van de week, dag van het jaar of week van het jaar*. Hierbij wordt vaak gerekend met het aantal dagen vanaf een zekere startdatum. Als de jaartelling niet meer klopt (een aantal jaren/maanden/weken/dagen valt over de eeuw-grens) zal het resultaat waarschijnlijk ook niet kloppen. En het jaar 2000 is wél een schrikkeljaar (zie hieronder). Bovendien is er de complicerende factor dat een aantal systemen de weekdag berekent vanaf een startdatum 1/1/00, er impliciet van uitgaande dat 00 staat voor 1900, en ook hierdoor verkeerde weekdagen berekent.

Een geval apart zijn de obscure berekeningen als $\text{VariabeleA} = 10000.01 * \text{VariabeleB}$ (zie tabel 1). Het getal 1461 is ook zo'n waarde die onopvallend is, maar wel met data te maken heeft. En '28' is het aantal jaar na welk de weekdagen en data weer gelijklopen. En zo zijn er nog wel meer waarden die in berekeningen op het oog niets bijzonders doen, maar wel geïnspecteerd moeten worden. Overigens duiden sommige berekeningen van tabel 1 er verwarrend genoeg op dat al wel eeuw-cijfers in gebruik zouden zijn.

Vergelijkingen

Met name bij het vergelijken van data kunnen veel interpretatiefouten optreden. Aan het jaartal '97' is niet te zien of het gaat om 1897 (een geboortedatum) of 1997 (een eventuele sterfdatum). En uw systemen zullen ook moeten begrijpen dat '10' een pensioneringsdatum is in de volgende eeuw. Kijkt u eens op uw creditcards. In de Verenigde Staten zijn er al geldautomaten geweest die die weigerden omdat het vervaljaartal '03' kleiner is dan het jaartal '96'.

Ook bij zaken als het al of niet pensioen- of stemgerechtigd zijn van een persoon worden – op het oog geheel juiste – vergelijkingen gemaakt. De crux is dat mensen direct op basis van de context zien dat bijvoorbeeld een leeftijd van -55 jaar niet in orde is, en er honderd jaar bij op zullen tellen. Computers zijn fundamenteel contextloos en de oplossing dat er in dergelijke situaties honderd jaar moet worden gecorrigeerd, zal expliciet moeten worden ingeprogrammeerd. De huidige Jaar 2000-problematiek komt er juist uit voort dat dat tot voor kort niet zinvol werd geacht.

Vergelijkingen van data komen ook terug in – soms in hoge mate geautomatiseerde – backup-procedures. Zonder extra aanpassingen zullen vele backup-tapes worden gevuld met verkeerde backups (ver-

keerde schema's), backup-tapes in een verkeerde volgorde worden vrijgegeven voor hergebruik en dergelijke. Gelieve er rekening mee te houden dat de millenniumwisseling een gebeurtenis is waarvoor juist bij uitstek backups nodig kunnen blijken te zijn van in potentie alle, maar dan ook alle elektronisch opgeslagen gegevens en programmatuur.

Dataformaten

Dataformaten zijn in wezen de bron van alle problemen. Door het slim gebruik van slechts twee posities voor een jaartal kunnen vele bytes aan opslagruimte worden bespaard. Tot voor kort gold dit als een punt van zorgvuldigheid, geheugen was tenslotte duur. Bovendien had niemand voorzien dat de toentertijd ontwikkelde programmatuur en data tot over de eeuwgrens in gebruik zouden zijn. Als echter ruimte moet worden gemaakt voor vier jaartalcijfers (of andersoortige opslag – zie paragraaf Oplossingen), kunnen fouten optreden van één of meer van de volgende soorten:

– *Overflow.* Indien bij een jaartal 99 precies 1 wordt opgeteld, zal wellicht zoals bedoeld een 00 als resultaat verschijnen, maar waar het honderdtal blijft is veelal onduidelijk. In sommige gevallen (welke is zeer moeilijk van tevoren te voorspellen) zal de werking van systemen daardoor in het ongereede raken, leidende tot onopvallende of groot-schalige rekenfouten of zelfs systeem-crashes.

– *Substrings.* In zeer veel gevallen wordt invoer als een string van karakters ingelezen en vervolgens met gebruikmaking van een substring-functie tot datumonderdelen geparset. Indien nu echter de invoer in de string verandert, zullen – afgezien van de lengte van de string, het invoerveld op het scherm, etc. – deze parse-functies moeten worden aangepast. Dit nog afgezien van het gebruik van een string als opslagformaat voor data. Vanwege de soms obscure variabelenamen zullen dus alle substring-functies moeten worden nagelopen.

– *Verschillende formaten.* Doordat bedrijfsbrede uniforme en consistent gebruikte datadefinities (data dictionary) een zeldzaamheid zijn, zullen de vele informatiesystemen die onderling gegevens met elkaar uitwisselen onderling verschillende formaten hanteren voor data. Waar nodig zullen vertaalslagen worden gemaakt in de interfaces. Die zullen echter aanpassing behoeven, bovendien zullen sommige oplossingen slechts tijdelijk zijn. Voordat een aangepast systeem operationeel kan worden, dienen de interfaces te zijn aangepast voor de overgangsfase. Zodra echter de andere systemen worden aangepast, dient zo'n 'verse' interface opnieuw te worden aangepast.

– *Uitwisseling met leveranciers en afnemers.* Het voornoemde interface-probleem speelt natuurlijk nog sterker zodra gegevensuitwisseling plaatsvindt met leveranciers en afnemers.

– *Scherm- en print-output.* Invoer- en uitvoervelden zijn mogelijk te klein om viercijferige in- en uitvoer te accepteren; door automatische carriage returns zal uitvoer chaotisch kunnen worden.

Data-opslag

Behalve in de verwerking kan er ook in het opslaan en ophalen van gegevens het een en ander mislopen. De voornaamste categorieën van acties met potentieel verkeerde resultaten zijn:

– *Sorteren.* '00' is kleiner dan '96' en of er nu oplopend of aflopend wordt gesorteerd, de resultaten komen niet overeen met wat de bedoeling was. Dit is één van de grootste problemen dat nu geldt. Indien op basis van de jaartalsortering beslissingen worden genomen, zullen die verkeerd zijn. Beslissingen op basis van First-In-First-Out geven de nieuwste elementen (met jaartal '01' bijvoorbeeld, in plaats van die met jaartal '96') voorrang, terwijl beslissingen op basis van Last-In-First-Out juist de oudste elementen eruit pikken. Bij een voorraad-

Beslissingen gebaseerd op jaartalsortering zullen verkeerd zijn.

systeem (bijvoorbeeld ook een bestand met pensioeringsdata) betekent zo'n selectie dat de meest verse voorraden als bedorven worden afgevoerd, terwijl voorraden die op het punt staan te bederven of reeds bedorven zijn, nog een fors aantal jaar in opslag zouden blijven.

– *Zoeken.* Bij het zoeken van een jaartal als '96' in een serie datumvelden kan men ervan uitgaan dat alleen de relevante jaartallen worden gevonden. Het zoeken op '01' echter zal een zeer groot aantal verkeerde resultaten opleveren omdat nu plotse-ling ook alle data met de eerste dag van een willekeurige maand of de eerste maand van een willekeurig jaar worden gevonden.

– *Indiceren in files of tabellen.* Grote bestanden met vele kleine elementen gebruiken vaak indices met boomstructuren. Indien naast grote jaartallen als 96, 97 ... nu ook kleinere als 00, 01, ... voorkomen, is niet zonder meer zeker dat de resulterende logica in de indices blijft zoals die bedoeld was.

Datumveld met speciaal gebruik

In het verleden zijn, niet altijd vanwege nalatigheid maar ook vanwege de mogelijkheid te besparen op opslagkosten, datumvelden en -variabelen in databases en programmatuur nogal eens oneigenlijk gebruikt om vlaggen, indicatoren en dergelijke in op te slaan. De logica die met dit soort misbruik werd geïmplementeerd, is nu praktisch alleen nog met nauwkeurige inspectie van de broncode herleidbaar.

Daarnaast zijn ook in datumvelden zeer vaak de waarden '99' – en terugtellend ook wel '98', '97', etc. – en '00' gebruikt ter indicatie van specifieke situaties. '99' duidde dan bijvoorbeeld op 'eeuwigdurend', terwijl '00' vrijwel altijd een verstekwaarde was en/of duidde op 'onbekend'. Nu deze getallen echter als jaartallen zinvol kunnen worden geïnterpreteerd, zal al dit soort misbruik moeten worden ongedaan gemaakt. Dit betekent niet al-

leen dat de gebruikers moeten worden geïnstrueerd in het vervolg andere codes te gebruiken (welke??) en de broncode moet worden aangepast (verstekwaarde, en de logica moet testen op de nieuwe coderingen), maar ook dat alle oude bestanden moeten worden doorlopen en eventueel aangepast.

Schrikkeljaar

Het jaar 2000 is een schrikkeljaar! Het jaartal is deelbaar door vier maar bovendien is het jaartal deelbaar door 400 en daarmee vervalt de eeuwscorrectie.

In veel gevallen is het ogenschijnlijk niet relevant of er al of niet een 29 februari 2000 is. Maar in feite gebruiken vrijwel *alle* berekeningen waarbij een datum betrokken is, op een of andere wijze het aantal dagen in een jaar of de dag van de week. En dus is het zaak om bij alle berekeningen te onderzoeken of er datumvelden worden gebruikt die een waarde na 28 februari 2000 (zouden moeten) kunnen bevatten, of dat het – reeds verstreken of totaal – aantal dagen van 2000 in de berekening wordt gebruikt, of dat een dag van de week na 28 februari wordt gebruikt.

Getroffen objecten

Op de vraag *wáár* het verkeerd gaat met de data in systemen kan op voorhand alleen gesteld worden, dat *álle* elementen van alle informatiesystemen (van mainframe-job tot spreadsheet) onderzocht moeten worden, dus:

- mensen en procedures;
- maatwerk;
- pakketten;
- talen;
- middleware;
- operating-systemen;
- tijd- en datumservices;
- hardware;
- data en databases;
- interfaces.

Hieronder zullen deze elementen worden besproken in het licht van hun specifieke problemen.

Mensen en procedures

In veel gevallen zullen medewerkers expliciet geïnstrueerd moeten worden dat datumvelden op invoerschermen hoe dan ook moeten worden ingevuld. Het niet invoeren zal de impliciete verstekwaarde '00' laten staan. Ook liet gebruik van '99' om het ontbreken van een jaartal aan te duiden of om 'doorlopend' te kunnen invoeren, zal moeten veranderen. Daarnaast moet niet vergeten worden dat, hoe goed een bedrijf de Jaar 2000-problematiek vooraf ook oplost, de mogelijkheid van het optreden van fouten zeker niet is uit te sluiten. Daarom zullen de medewerkers op enigerlei wijze gestimuleerd moeten worden om te zijner tijd (en reeds nu al, als er data over de eeuwsgrens heen worden gebruikt) alert te zijn op het optreden van fouten.

Een andere factor die hier een grote rol speelt, is die van het werven, trainen en gemotiveerd houden van de (grote scharen) programmeurs en managers

die nodig zijn om het probleem op te lossen. Hun werk levert in principe weinig meerwaarde voor het bedrijf; het voegt op zich geen gebruikersfunctionaliteit toe terwijl ze hoe dan ook wel impliciet of expliciet worden aangekeken op hun hoge kostenbudgetten. Voeg daarbij dat vrijwel alle prognoses nog steeds wijzen op een enorm tekort aan programmeurs en andere informatici, en dat de dreiging van vertrek wegens een hoger salaris elders ook al steeds groter wordt. Qua motivatie is het ook niet bevorderlijk dat alle betrokkenen zeer wel in de gaten hebben dat er na 1 januari 2000 heel wat minder werk te doen zal zijn en er dus veel overtolligheid dreigt. Niet echt redenen om oog te hebben voor maximale kwaliteit. Het belang van het geheel voor het voortbestaan van het bedrijf vraagt daar echter wel om en dus zal het betrokken personeel sterk gemotiveerd moeten worden.

Maatwerk

Bij maatwerk speelt het ontbreken van adequate documentatie vaak een grote rol. Vaak betreft maatwerk (onderdelen van) de software die nu net specifiek voor een bedrijf is, die soms een bedrijf maakt tot wat het is. Hierdoor is kennis over het maatwerk zelden elders te halen.

Bovendien zijn de grotere maatwerksystemen typische *legacy*-applicaties die toch al niet bekend stonden om hun onderhoudbaarheid en die anderzijds zo oud zijn dat bij het bouwen ervan het jaar 2000 ver buiten beeld was. Hier geldt vaak: 'If it works, don't fix it!', maar voor het Jaar 2000-bestendig maken zal er in de code *moeten* worden ingegrepen. Helaas blijkt juist ook bij maatwerk vaak dat zelfs de broncode niet meer beschikbaar is. 'Reverse engineering' kan dan in sommige gevallen nog wel wat helpen – al zullen heldere variabelenamen als DatumInDienst zeldzaam zijn –, maar ook dit trekt een zware wissel op de beschikbare mensen en hulpmiddelen.

Eigenlijk zou dit soort applicaties bij voorkeur van de grond af aan herbouwd moeten worden. Hoewel daar kennelijk tot nu toe geen prioriteit voor te krijgen was en de Jaar 2000-problematiek alleen maar meer werk betekent, is de noodzaak van herbouw in dit soort gevallen sterk toegenomen.

Batch-jobs, geschreven in een Job Control Language, zijn ook vaak maatwerk. Ze moeten beslist niet vergeten worden bij systeemveranderingen. Niet alleen zouden later nog wel eens fouten kunnen opduiken doordat de in de batch-jobs genoemde programma's niet meer bestaan (door vervanging door nieuwe), ook kunnen de batch-jobs zelf datum- of dag-van-de-week-afhankelijke elementen bevatten en dus gevoelig zijn voor de Jaar 2000-problematiek.

Pakketten

Geeft maatwerk problemen door een gebrek aan documentatie en kennis, bij pakketten zou het oplossen van die gebreken als het ware gedelegeerd zijn aan de – naar verwachting betrouwbare – leveranciers. Hier blijken toch wel twee problemen aan vast te zitten:

- De leveranciers hebben bepaald niet in alle gevallen de documentatie of kennis zelf nog in huis.

Voorzover de leveranciers dan bereid zijn aanpassingen alsnog door te voeren, betekent dat natuurlijk op voorhand een aanzienlijke bedreiging voor de kwaliteit van het werk, nog afgezien van de tarieven die leveranciers zullen (gaan) rekenen. Pakketten waarvoor de ondersteuning in het verleden al is stopgezet, kunnen vrijwel altijd eigenlijk alleen nog maar integraal worden vervangen.

– De leveranciers vallen buiten de ‘span of control’ van de organisatie. Hierdoor is niet afdwingbaar en vaak alleen al moeilijk verifieerbaar dat de leveranciers de oplossingen goed (kwalitatief voldoende) implementeren, en daarbij de oplossingen (juiste formaten, uniformiteit) kiezen die het meest geschikt zijn voor uw organisatie. Bovendien bent u afhankelijk van de leveranciers om de verbeterde versies van de pakketten tijdig opgeleverd te krijgen. Wel kan zekerheid over de kwaliteit van de verbeteringen worden verkregen door verklaringen van de leverancier of, onafhankelijk, van een EDP-auditor.

Vervolgens zal in de organisatie dan nog een traject van installatie en controle moeten worden doorlopen. Ten opzichte van het zelf moeten wijzigen van maatwerk zal de inspanning voor dat traject echter meevallen.

Talen

Vele programmeertalen bieden geen ruimte om datumvelden met vier jaarcijfers te gebruiken. Hoewel dit probleem relatief eenvoudig is op te lossen met de aanschaf van compilers en dergelijke die een nieuwe versie van de taal met vier jaarcijfers aankunnen of door bruggen te gebruiken (zie Oplossingen), betekent het wel een complicerende factor. Wat te doen als zo’n nieuwe versie van de taal niet voldoet of niet eens bestaat? Wat als bruggen een te omslachtige oplossing zijn?

Een aan talen gerelateerd probleem is dat, zoals hiervoor genoemd werd, zowel maatwerk als pakketten nogal eens slecht gedocumenteerd zijn en de kennis en expertise dun gezaaid. In de praktijk blijken de minst gedocumenteerde systemen niet alleen vaak de belangrijkste voor de organisatie (ze zijn vaak de systemen waar het bedrijf op draait en/of die unieke functionaliteit bevatten), maar bovendien zijn ze nu juist de systemen die zijn geprogrammeerd in de minst verbrede en meest in onbruik geraakte programmeertalen. Onnodig te zeggen dat wijzigingen van de programmatuur daardoor grote (financiële) inspanningen zullen vergen.

Middleware

Middleware, de software die netwerken transparant maakt voor applicaties, is de laatste jaren sterk naar voren gekomen in de softwareportefeuille. Ook in middleware is een fors aantal datumbewerkingen verwerkt. Men denke alleen al aan de vele mechanismen voor berichtauthenticatie waarbij berichten gecodeerd of gelabeld worden met datum/tijd-afhankelijke data.

Een klasse apart zijn daarbij de tijd- en datumservices. Deze maken vaak onderdeel uit van de middleware en hebben juist vrijwel als enige doel systeemtijden en -data te manipuleren, bijvoorbeeld

om die in een netwerk van systemen gelijk te schakelen. Enerzijds is het een voordeel dat zo veel datumfuncties in aparte stukken middleware geconcentreerd zijn, anderzijds worden deze functies juist vaak van externe leveranciers verkregen met alle afhankelijkheid van dien.

In het belang van het voortbestaan van het bedrijf zal het betrokken personeel sterk gemotiveerd moeten worden.

Operating-systemen

De fouten die in operating-systemen kunnen optreden zijn:

- een systeemklok die de fout ingaat. Bij sommige operating-systemen springt de interne klok te zijner tijd op 0, waarbij 0 een willekeurige startdatum kan aangeven;
- operating-systeem-eigen database-managementsystemen die van slag raken. Zie Data en databases hieronder;
- wijzigingsdata van files en versiebeheer ervan die in het ongerede raken door foutieve dataaanduidingen of sorteringen op datum;
- application programmable interfaces (API's) naar het operating-systeem die een viercijferig jaartal niet aankunnen.

Juist bij operating-systemen zijn er leveranciers die beweren dat hun systeem Jaar 2000-bestand zou zijn, terwijl in praktijktests blijkt dat er toch nog datumfuncties zijn die foutieve resultaten geven. Anderzijds willen er nogal eens geruchten de ronde doen over foutieve resultaten die niet hard te maken zijn. Zelf testen lijkt hiervoor de enige reële oplossing.

Hardware

Bovengenoemde problemen van operating-systemen gelden tevens voor hardware c.q. microcode. Nogal wat hardwaresystemen zullen overigens al eerder dan op 1 januari 2000 problemen krijgen omdat de interne-kloktellers op 0 springen boven een zekere waarde en vanaf daar weer verder gaan.

De hardwareproblemen zullen met name kunnen optreden in systemen waarin de realtime-elementen een belangrijke rol vervullen, zoals GSM-telefoons en hartbewakingsapparatuur. Een bekend voorbeeld is het Global Positioning System, waarvan de interne tellers al in augustus 1998 door 0 zullen gaan en de geografische-positieresultaten onzinnig zullen zijn. Ook telefooncentrales en liften kunnen na 1 januari 2000 chaotisch gedrag gaan vertonen. Gebouwbeveiligingssystemen zullen met directe ingang iedereen – dus ook een eventuele ‘rampenbrigade’ – de toegang ontzeggen omdat alle permits verlopen zijn.

Data en databases

Het Jaar 2000-probleem is ontstaan door de besparingen in opslagruimte. Nu worden die besparingen dus potentieel ongedaan gemaakt. Vergeet

daarbij niet dat zodra de organisatie te zijner tijd een inschatting heeft kunnen maken van de omvang van eventueel extra benodigde opslagruimte, de markt voor opslagmedia al flink op toeren zal zijn met bijbehorende prijzen en levertijden.

Fouten kunnen zowel optreden in de gegevens zelf als in de database-managementsystemen:

- In de gegevens zelf zullen veelal de eeuwaanduidingen ontbreken. Als er geen specifieke actie wordt ondernomen zou aangepaste programmatuur die data wel eens aan kunnen zien voor data uit de eerste eeuw ... De gedachte om zonder meer alle datafiles door te worstelen en alle jaartallen te voorzien van twee eeuwcijfers klinkt aantrekkelijk maar brengt vanwege opslagruimteproblemen, indexeringen, referentiële integriteit en dergelijke nog behoorlijk wat problemen met zich mee.

Luid en duidelijk reclame maken voor de Jaar 2000- projecten zal hard nodig zijn.

- De database-managementsystemen (DBMS'en) zijn soms onderdeel van het operating-systeem (zie aldaar). De DBMS'en die als zelfstandige applicaties (pakketten) draaien, kennen ook de daarbij behorende problemen. Met name bij DBMS'en zal het tevens zo zijn dat het te gebruiken dialect mogelijk te nauwe restricties oplegt aan de mogelijke datumformaten, het DBMS een aantal mogelijk verkeerd lopende sorteringen uitvoert of verkeerd lopende indexeringen hanteert.

Interfaces

Buiten simpele kantoorautomatisering komen standalone-applicaties steeds minder voor. Dit betekent een toename van de interrelaties tussen systemen. Was het daarbij al steeds moeilijk om tot stabiele definities en implementaties van die interfaces te komen, zodra zo dadelijk voor het oplossen van de Jaar 2000-problematiek vele of alle systemen en de interfaces weer met verschillende snelheden worden aangepast, zal de interface-problematiek weer in volle omvang terugkomen.

Zodra de invoerformaten veranderen zou de interface die mogelijk niet meer kunnen accepteren doordat de invoer niet meer door de invoercontroles komt. In de transformatie kunnen dan vervolgens bij invoer die wel door het invoerfilter komt, rekenfouten optreden doordat de verkeerde berekeningen en/of vergelijkingen spaak lopen (zie boven). Als de uitvoerformaten niet tijdig worden aangepast, zullen de uitvoerstromen mogelijk door de ontvangende applicaties worden geweigerd. En als de interface goed lijkt te werken, kunnen de transformaties in semantische zin nog wel eens niet-integere informatie opleveren.

Inzake de Jaar 2000-bestendigheid kunnen interfaces wellicht het beste worden beschouwd als vertaalapplicaties. De problemen zijn dan te herleiden tot de invoer, de transformatie en de uitvoer. Invoer- en uitvoerformaten zullen veranderen als ge-

volg van veranderende toeleverende en afnemende applicaties en de transformatie zal veranderen (per interface in principe tweemaal: zowel bij wijziging van de invoerformaten alsook bij de wijziging van de uitvoerformaten) om de integriteit van de overdracht te waarborgen.

Voorzover de interfaces werken tussen applicaties binnen de organisatie zelf, zijn de problemen nog te overzien. In principe bevinden de applicaties, en daarmee de definities van invoer- en uitvoerformaten, zich binnen de 'span of control' van een functionaris binnen de eigen organisatie. Daardoor kunnen de aanpassingen op elkaar worden afgestemd – inhoudelijk met behulp van bijvoorbeeld een centrale datadefinitie en qua tijdigheid – en kan de kwaliteitsbewaking in lijn worden gebracht met de organisatiestandaarden.

Zodra de interfaces communiceren met applicaties buiten de eigen organisatie ontstaan echter besturingsproblemen. In het verleden is het al ondoenlijk gebleken wijzigingen in applicaties (waaronder interfaces) altijd op de afgesproken tijd en met voldoende kwaliteit op te leveren. Als de organisatie echter op deze wijze afhankelijk wordt van andere, doordat men rekent op een ongestoorde samenwerking tussen de applicaties waarvan men afhankelijk is geworden, zullen de verantwoordelijken heel wat minder rustig slapen.

Het is daarbij te hopen dat de organisatie en haar leveranciers en afnemers bilateraal of multilateraal tot afspraken kunnen komen omtrent de dataformaten, en dat voor zowel de overgangsfase als de definitieve situatie. Dit punt zou een sterke motor moeten zijn om aan te sluiten bij internationale standaarden.

Juridische aspecten

Zodra er hoge kosten met weinig baten in het spel zijn in combinatie met menselijk handelen, komt al gemakkelijk de kwestie van aansprakelijkheid naar voren. De leverancier van software of data had immers tijdig voorzorgen kunnen nemen opdat zijn producten over de eeuwsgrens heen op hun normale manier blijven functioneren. Nu dat niet gebeurd is, zal die leverancier dan toch aansprakelijk zijn?

In veel gevallen zal een beroep van de leverancier op overmacht niet snel gegrond zijn. De millenniumwisseling was ruim van tevoren bekend en er was (!) voldoende tijd om tegenmaatregelen te nemen. Er zijn ook bedrijven die al een jaar of vijf bezig zijn met het Jaar 2000-bestand maken van hun systemenportefeuille.

Wanprestatie en aansprakelijkheid

Toch zal een claim bij een leverancier niet makkelijk hard te maken zijn. Bij aflevering functioneerde het systeem naar behoren en naar goedkeuren van de afnemer. En de specificaties zeiden niets over Jaar 2000-bestendigheid, noch over het tegengaan van technische slijtage door het voortschrijden van de tijd. Bovendien was het gebruik van twee jaartallen gebruikelijk in de software-industrie, zo het al niet gold als positief kwaliteitskenmerk van-

wege de ermee te behalen kostenbesparingen ten gunste van de afnemer. Voor wat betreft de leverancier is daarmee de kous af. Slechts indien uw automatiseringscontract daarin voorziet, kunt u wanprestatie hard maken en tijdig wijzigingen afdwingen.

De genoemde uitzonderingen daargelaten begint pas het laatste jaar door te dringen dat Jaar 2000-bestendigheid een onderdeel van de specificaties van nieuwe systemen moet zijn. Prompt zullen leveranciers desgevraagd direct kunnen aangeven dat hun software geheel Jaar 2000-bestand is. Hier past echter een belangrijke waarschuwing. Als een leverancier zegt dat de software in orde is, biedt dat geen enkel bewijs bij later blijkende tekortkomingen. Het is al voorgekomen dat zogenaamd Jaar 2000-bestendige software niet door de tests kwam. Uw enige houvast is in zo'n geval een verklaring op papier. Van een leverancier mag best gevraagd worden dat zwart op wit verklaard wordt dat de software Jaar 2000-bestendig is. Als de leverancier meent wat hij zegt, kan dat geen probleem vormen, 'wat goed is, kan ook op papier'. Bij uitbestede dienstverlening zou bijvoorbeeld een 'Jaar 2000-paragraaf' in een service level agreement moeten worden opgenomen.

Vraagt u echter wel deskundig advies inzake de formulering van garanties en/of andere contractbepalingen, aansprakelijkheid voor softwareproblemen blijft immers een juridisch mijnenveld. En al hebt u de verklaring, dan nog is het verstandig zelf uitgebreid te testen. Want helpt het straks om excuses of een schadebedragje te ontvangen als een heel systeem (of, beslist niet denkbeeldig, zelfs uw hele bedrijf) is gecrasht? Ook andere juridische aspecten spelen een rol.

Licenties

Eén van de juridisch getinte onderwerpen zullen de systemen blijken te zijn die zichzelf bij het verlopen van de licenties automatisch uitschakelen. In nogal wat van die systemen worden data na 31 december 1999 zodanig verkeerd verwerkt dat die uitschakelmechanismen in werking treden. Terwijl de licentie nog gewoon doorloopt ... Het is van groot belang deze problematiek in de licentie-overeenkomsten tot uitdrukking te brengen.

Inzake fiscaal-juridische aspecten zullen overigens ook tijdig afspraken moeten worden gemaakt met de belastinginspecteur, omdat aanpassingen in systemen en gegevens op deze schaal het voldoen aan fiscale administratieve bewaarplichten in de knel kunnen brengen. De belastinginspecteur dient evenwel in redelijkheid met de belangen van de belastingplichtige rekening te houden.

OPLOSSINGEN

In technische zin zijn er twee methoden om oplossingen te implementeren: het wijzigen van procedures of het wijzigen van data.

Wijzigen van procedures

Het merendeel van de oplossingen is onder te brengen in de categorie procedurele wijzigingen, ofwel wijzigingen in de functionaliteit van de programmatuur. In een aantal gevallen zal het nodig blijken meerdere van deze oplossingen tegelijk in een module toe te passen.

Codering/compressie

Een mogelijkheid die wel wordt voorgesteld, is het coderen van jaartallen in bijvoorbeeld hexadecimale. Het jaar 1999 wordt dan het jaar '63', 2000 ('100') wordt dan '64'. Een voordeel van deze methode is dat berekeningen, sorteringen en dergelijke – mits ze worden aangepast voor hexadecimaal rekenen! – verder ongewijzigd kunnen blijven. Een groot nadeel is dat de resultaten slecht leesbaar zullen zijn zonder additionele presentatieroutines.

Conversie naar viercijferige jaartallen

Eén van de meest voor de hand liggende oplossingen is het expanderen van alle in gebruik zijnde jaartallen naar vier cijfers (zie Wijzigen van data hierna). Daarbij dienen dan wel allerlei aanpassingen aan programmatuur te worden verricht, zoals aanpassingen aan substrings, scherm- en print-lay-outs, interfaces, etc. Daarnaast: hardwaredata zijn niet te expanderen zonder ingreep in de hardware.

Fixed window

Een andere oplossing is die welke mensen contextgevoels al toepassen: het interpreteren van bijvoorbeeld jaartallen 50 tot en met 99 als zijnde van de twintigste eeuw, en jaartallen van 0 tot en met 49 als zijnde van de volgende. Hierbij wordt dan dus een window (interval) gehanteerd van honderd jaar, vastgelegd op 1950 tot en met 2050. Berekeningen en vergelijkingen met jaartallen zullen dan telkens moeten worden gecorrigeerd door het optellen van 50 en modulo 100 om zinvolle jaartallen te krijgen.

Daarbij dient er wel voor gewaakt te worden in de diverse systemen verschillende window-grenzen te hanteren. Als de ene applicatie '40' begrijpt als zijnde 2040 (de window-grens ligt bij 50) kan een andere applicatie die van dezelfde gegevens gebruik maakt dat interpreteren als 1950 (window-grens bij 30). Niet alleen zullen de verschillende grenzen lokaal optimaal zijn gekozen en is een organisatiestandaard dus niet lokaal optimaal, ook zullen de window-grenzen in interfaces naar andere organisaties afhankelijk zijn van wat die andere organisaties hanteren.

Sliding window

Een variant op het fixed window is het idee van een sliding window, waarbij de grenzen meeschuiven met het voortschrijden van de jaren. Het voordeel daarvan is dat als de huidige situatie het nodig maakt dat bijvoorbeeld én 1960 én 2059 correct worden herkend, dat volgend jaar zeer waarschijnlijk 1961 en 2060 zijn. De window-grenzen worden aldus geparametriseerd. Dit betekent een (hopelijk) doorlopend passend window, maar anderzijds een toename van de complexiteit. De attentiepunten van het fixed window gelden ook hier.

Gemeenschappelijke datum/tijd-routine

Een andere mogelijkheid is het gebruik van één of een beperkt aantal gecentraliseerde datum/tijd-routines. Hiermee wordt een grote mate van standaardisatie bereikt en een concentratie van onderhoudsinspanningen – mits alle programmatuur deze routines op de juiste manier gebruikt! Bovendien creëert deze oplossing natuurlijk een zeer groot aantal aanroepen van die routines met de gevolgen van dien voor de prestaties van de systemen.

Interfaces: bruggen

De interfaces tussen applicaties kunnen en moeten vaak, tijdelijk of permanent, worden voorzien van zogenaamde bruggen: aparte softwaremodules die de vertaalslag verzorgen tussen de in beide applicaties gebruikte oplossingen. Deze bruggen, in de praktijk bestaande uit veel kleine modules tussen de diverse systemen, vergen nogal wat onderhoud. Ze zijn namelijk gevoelig voor (formaat-)wijzigingen aan zowel de invoer- als de uitvoerkant.

Systeemklokken terugzetten?

Het simpelweg terugzetten van systeemklokken lijkt een eenvoudige oplossing, lost echter niets op maar brengt juist een grote hoeveelheid problemen met zich mee, bijvoorbeeld:

- bij elke systeemherstart moet zorgvuldig worden gecontroleerd dat de systeemdatum goed staat;
- alle interfaces naar andere organisaties moeten worden aangepast om een juiste vertaling te maken;
- alle invoer en uitvoer moet worden vertaald naar de juiste data;
- alle berekeningen en vergelijkingen moeten worden gecontroleerd op juistheid van de berekening;
- alle datumgegevens moeten worden nagezien: (historische) data tussen de 'systeemdatum' en nu worden verkeerd geïnterpreteerd;

- elk nieuw systeem zal terugwaarts moeten worden aangepast om in het 'klok terug'-stramien te passen;
- etc., etc.

Vervangen van systemen

De meest drastische oplossing, echter nogal eens de meest effectieve, is het geheel vervangen van systemen door nieuwe, Jaar 2000-bestendige systemen. Voorzover applicaties relatief zelfstandig functioneren, kan hierdoor een aanzienlijke besparing worden bereikt door afstand te doen van oude spaghetti-code. Anderzijds zullen de interfaces met andere systemen en de historische data toch moeten worden aangepast. Het vervangen van alle systemen zal overigens soms onmogelijk zijn door een te groot aantal systemen of onrendabel zijn omdat in de oude systemen relatief weinig (goedkoop) hoeft te worden gewijzigd. Wel kan de vervanging als elk ander grootschalig turnkey-project worden beheerst.

Wijzigen van data

Een geheel andere oplossingsmethode is het wijzigen van alle door een systeem gebruikte data door toevoeging van twee eeuwcijfers aan alle voorkomende jaartallen. Dit is een relatief eenvoudige oplossing, maar vergt toch een behoorlijke inspanning. Niet alleen blijft het lokaliseren van databerekeningen nodig (zoals voor invoercontroles, interfaces en scherm- en printformaten), maar ook zullen bruggen nodig blijven naar applicaties die nog niet zijn aangepast of die om welke reden dan ook zijn aangepast met de procedurele methode. Dit nog afgezien van het moeten aanpassen van de enorme hoeveelheden historische data.

Tabel 2 geeft een overzicht van de voor- en nadelen van beide aanpakken. Een gedegen analyse van de kosten zal moeten bepalen welke methode de voorkeur krijgt.

Tabel 2. Procedurele versus datagerichte oplossingen.

| Wijzigen procedures | Wijzigen data |
|--|--|
| <p><i>Voordelen:</i></p> <ul style="list-style-type: none"> – Programmatuurwijzigingen kunnen in relatief kleine stukken en relatief geïsoleerd worden doorgevoerd en getest. – Alleen kleine stukken software moeten worden aangepast. De functionaliteit van de systemen blijft onaangetaast. – Bestaande (ook gearchiveerde) data behoeven geen aanpassing. <p><i>Nadelen:</i></p> <ul style="list-style-type: none"> – De logica kan complex worden, ook leidende tot hogere onderhoudskosten. – Controles op eeuwen zijn context-specifiek. – Als steeds meer interfaces met vier cijfers werken neemt de noodzaak voor bruggen toe. – De logica beïnvloedt de prestaties van systemen. – De complexiteit van sorteerroutines zal toenemen. | <p><i>Voordelen:</i></p> <ul style="list-style-type: none"> – Broncodewijzigingen zullen eenvoudiger blijken te zijn. – Heft ambiguïteiten en volgordeproblemen op. – Elimineert de overhead van het documenteren van complexe logica. – Als steeds meer systemen viercijferige interfaces krijgen, neemt de behoefte aan brugfuncties af. – Maakt een consistente aanpak mogelijk. <p><i>Nadelen:</i></p> <ul style="list-style-type: none"> – Het wijzigen van datastructuren moet in veel systeem-elementen (programmatuur, softwarebibliotheken) worden doorgevoerd en alle software zal waarschijnlijk moeten worden gehercompileerd. – Gearchiveerde data moeten ook gewijzigd worden. – Toenemende vereisten aan de opslagcapaciteit. – Er moeten naar verhouding veel relatief grote tests worden uitgevoerd. |

ORGANISATORISCHE ASPECTEN

Alle organisaties die niet reeds lang met het Jaar 2000-probleem bezig zijn, zouden een 'sense of urgency' moeten voelen en direct aan de slag moeten. Een adequate oplossing van het probleem is echter ook in belangrijke mate een kwestie van een zorgvuldige aanpak van de organisatorische aspecten. Daarbij speelt een groot aantal zaken, die ingedeeld kunnen worden naar de volgende items:

- strategie;
- totaalplanning;
- analyse;
- wijzigingen;
- conversie;
- testen;
- migratie;
- evaluatie.

Hieronder wordt een aantal aspecten toegelicht waarmee ongeacht de specifieke, in een organisatie bruikbare projectmanagementmethodieken rekening moet worden gehouden.

Strategie

Bij het bepalen van de algemene strategie waarmee de Jaar 2000-problemen worden aangepakt, dient met name aandacht te worden aan het kiezen van voorkeursoplossingen, de reikwijdte en het onderzoeken van de haalbaarheid van de tijdi-ge implementatie.

Voorkeursoplossingen

Er zal reeds in een vroeg stadium moeten worden gekozen voor de voorkeursoplossingen voor het Jaar 2000-bestendig maken van alle applicaties. De mogelijkheden zijn:

- applicaties aanpassen en data zo veel mogelijk ongemoeid laten;
- data aanpassen en applicaties zo veel mogelijk ongemoeid laten;
- nieuwe applicaties kiezen en de data zo nodig daaraan aanpassen.

Hierbij dienen dan tevens de voorkeuren voor de technische oplossingen – waaronder de te hanteren kwaliteits- en dataformaat-standaarden – te worden bepaald. Met deze twee beslissingen kan het werk zo veel mogelijk gestandaardiseerd worden, zodat op de aspecten projectmanagement, kennis en ervaring en tool-ondersteuning zo veel mogelijk economies of scale kunnen worden behaald, en bovendien het aantal aan te passen interfaces tussen systemen zo beperkt mogelijk blijft.

Overigens dient men er wel voor te waken te rigide vast te houden aan de voorkeursoplossing. De aangepaste (of nieuwe) systemen moeten ten minste zo goed functioneren als de oude, het kan niet zo zijn dat omwille van de standaardisatie het aangepaste systeem qua functionaliteit of prestaties (veel) slechter functioneert dan voorheen.

Alles ineens of alleen Jaar 2000

Bij de strategiebepaling moet ook worden beslist of bij het aanpassen van systemen ook allerlei andere

wijzigingen zoals toevallig gevonden fouten of aanpassingen voor de Euro-introductie binnen dezelfde (deel)projecten worden aangepakt. Vanuit de praktijkervaringen zijn de meningen verdeeld, met een voorkeur voor het *niet* meenemen van additionele wijzigingen. De problematiek is al zo omvangrijk en complex en al dat soort extra's zal de complexiteit alleen maar vergroten.

Het verdient aanbeveling in het Jaar 2000-project geen additionele wijzigingen mee te nemen.

Het is zeker af te raden om behalve preventief onderhoud ook nog uitbreidingen van de functionaliteit door te voeren. Dit op zich maakt namelijk het oplossen van de Jaar 2000-problemen vrijwel altijd al onbeheersbaar.

Haalbaarheid

De haalbaarheid van alle benodigde of gewenste aanpassingen moet ook reeds in een zo vroeg mogelijk stadium worden bepaald. Er zijn namelijk enkele moeilijk oplosbare knelpunten:

- *De beschikbare tijd.* Die is minder dan drie jaar! Anders dan bijvoorbeeld bij de Euro-problematiek is de deadline hard als graniet: 1 januari 2000. En voor veel systeemaanpassingen zal de deadline zelfs veel eerder liggen; denk aan het reeds voorkomen van data over de eeuwgrens heen in huidige systemen. Het niet halen van de deadline heeft aanzienlijke consequenties, van gezichtsverlies tot faillissement. En dat laatste is beslist geen hypothetische aanname.

- *De beschikbare middelen.* De organisatie heeft budgetten die maar net toereikend zullen zijn voor de normale gang van zaken, en een Jaar 2000-project zuigt (onder andere) zeer veel middelen (budget, maar ook computercapaciteit) op. Daar het project – als het goed is – in principe *geen* geld of uitgebreide functionaliteit oplevert, zal het zeer moeilijk zijn de benodigde grote extra budgetten te verkrijgen.

Er zijn schattingen die zeggen dat één op elke vijftig programmaregels een datumverwijzing bevat die geïnspecteerd moet worden. Gezien de vele miljoenen zo niet veel meer programmaregels die tegenwoordig bij ieder bedrijf in gebruik zijn, is de gedachte dat er nog tijd genoeg is dus hoe dan ook onjuist.

Schattingen van de integrale kostprijs van wijzigingen gaan al uit van prijzen in de orde van een Euro per aangepaste regel. De opbrengsten van allerlei aanpassingen zijn daarentegen nihil, de software zal – als het goed is – alleen maar de functionaliteit behouden die hij al had. Voorwaar een probleem om dan voldoende budgetten te verkrijgen ...

- *De beschikbare menskracht.* Hiervoor geldt hetzelfde als voor de middelen: de beschikbare mensen zullen al druk bezet zijn (zeker de experts). Bovendien geldt dat de arbeidsmarkt voor informatici

steeds meer trekken van overspannenheid begint te vertonen, hetgeen betekent dat een toereikend aantal kwalitatief voldoende (en ter zake kundige) medewerkers al steeds moeilijker tegen aanvaardbare tarieven is in te huren.

– Vorenstaande punten resulteren in een relatief zeer grote kostenpost. Hoe langer een organisatie echter wacht met de grootschalige aanpak van het probleem, des te moeilijker (duurder) zal het worden tijdig oplossingen te implementeren.

Totaalplanning

De planningsaspecten van Jaar 2000-projecten verschillen niet principieel van die van andere automatiseringsprojecten, en zullen hier dan ook niet in extenso worden uiteengezet. Wel is een aantal specifieke kanttekeningen te plaatsen.

Doelen stellen

Een eerste stap, die nogal eens wordt overgeslagen, is dat de doelen van de hele operatie moeten worden vastgesteld. Zonder deze doelen wordt het moeilijk te bepalen of de organisatie (ver genoeg) op de goede weg is. Tevens kunnen de globale doelen worden vertaald in doelen, deadlines en afhankelijkheden voor deelprojecten, zodat die beter gericht kunnen worden op de globale doelen die anders impliciet zouden blijven. Zonder globale doelen ontaarden de vele projecten in een stuurloze kluwen, terwijl in dit geval de doelen voor zo veel projecten nu juist wel onder een gemeenschappelijke noemer te brengen zijn.

De resultaten in sleutelvelden zullen extra goed moeten worden gecontroleerd.

Naast het vaststellen van de doelen dienen ze natuurlijk ook gecommuniceerd te worden; zonder communicatie zijn doelen nutteloos.

Taken definiëren

Ook hierbij speelt het inrichten van een 'concerted effort' een belangrijke rol. De Jaar 2000-problematiek laat geen ruimte voor het pas laat ontdekken van omissies van taakgebieden. Tevens biedt het tijdig zo volledig mogelijk definiëren van de taken de mogelijkheid de bemensing naar behoren in te plannen.

Inventarisatie

De problematiek dwingt de organisatie om – nu dan eindelijk – een volledige inventarisatie te maken van alle operationele en op stapel staande (!) systemen. Het is welhaast onvermijdelijk dat de applicaties die hierin vergeten worden, Jaar 2000-problemen zullen hebben (Murphy).

De inventarisatie kan niet alleen bestaan uit het simpelweg opsommen van de systemen; ook zal direct moeten worden gezocht naar broncodes, documentatie, etc., en zal een eerste inschatting moe-

ten worden gemaakt van bijvoorbeeld de omvang van de code, de gevoeligheid voor Jaar 2000-problemen en de complexiteit van de oplossingen.

Tevens kan de inventarisatie bijdragen tot het indelen van de systemen in onderling samenhangende clusters. Door de clusters zó te definiëren dat ertussen relatief weinig systeemkoppelingen zijn, kan het aantal aan te passen interfaces en te ontwikkelen bruggen worden beperkt. Criteria voor clustering kunnen onder andere zijn:

- gelijke prioriteit die van doorslaggevend belang is voor het overeind blijven van de organisatie;
- minimalisering of vereenvoudiging van het benodigde aantal en/of de complexiteit van de interfaces;
- onderlinge standaardisatie van oplossingen;
- onderlinge uniformiteit van programmastructuren (vereenvoudiging van de implementatie);
- groepering rond gebruikersgroepen.

Demonstratieproject

Vrij snel na aanvang van de aanpak van het Jaar 2000-probleem zal een demonstratieproject moeten worden gestart. Hiermee kan niet alleen het bewustzijn voor het probleem worden versterkt, tevens kan ervaring worden opgedaan in het oplossen ervan. Bij de keuze van het demonstratieproject zal een afweging moeten worden gemaakt tussen urgentie, complexiteit en slaagkans.

Als te veel nadruk wordt gelegd op de slaagkans alleen, zal ongetwijfeld worden gekozen voor een relatief onbelangrijk systeem waardoor de invloed van de Jaar 2000-problemen op de organisatie (en als gevolg daarvan de bereidheid voldoende budgetten beschikbaar te stellen) gemakkelijk onderschat zal worden. Bovendien is dan weliswaar ervaring opgedaan, maar die zal veel beperkter en minder breed zijn dan nodig is voor grotere en complexere systemen.

Prioriteitstelling

Vanwege de omvang van de problemen zal er zeer beslist een prioriteitstelling moeten worden gemaakt. Daarbij zal het belang van een systeem voor de bedrijfsuitoefening veruit het belangrijkste criterium dienen te zijn.

Inzake de prioriteitstelling wordt steeds meer de term triage gebruikt. Hiermee wordt in militaire kringen de prioriteitstelling aangeduid die een arts van een veldhospitaal moet toepassen als een (te) groot aantal gewonden wordt binnengebracht: het heeft geen zin dan tijd te verspillen aan kleine pijntjes of hoogst risicovolle operaties, veeleer gaat het erom de prioriteiten te stellen om te redden wat er te redden valt. Dat deze term in zwang begint te raken inzake Jaar 2000-portefeulles is tekenend voor de stijgende spanning.

De prioriteitstelling dient natuurlijk op een zinvolle wijze te worden gekoppeld aan de eerder gemaakte inschattingen van de benodigde inspanning per systeem en de clustering opdat de beschikbare capaciteiten optimaal worden benut. Indien echter blijkt dat een essentieel systeem on-

mogelijk met de (strategische) voorkeursmethode kan worden aangepast, dienen alternatieve oplossingen te worden bedacht en dient de volgorde in de projectenportefeuille te worden heroverwogen.

In tabel 3 zijn enkele speciale data aangegeven waarmee rekening zal moeten worden gehouden bij de prioriteitstelling.

Mijlpalen

Het vastleggen van mijlpalen is nodig omdat uitsstel en afstel niet alleen hinderlijk maar zelfs fataal kunnen zijn. Het niet halen van mijlpalen zal al snel noodgrepen nodig maken; de plannings zullen ongetwijfeld geen speelruimte bieden.

Data-administratie

Ter bevordering van de efficiëntie van de oplossingen zullen centrale data-administraties moeten worden opgezet, waarin in principe alle datatypes van alle systemen staan geregistreerd. Standaardisatie van oplossingen met navenant verminderde noodzaak van (blijvende) interface-aanpassingen wordt hierdoor gemakkelijker. In het geval een centrale data-administratie voorheen wellicht prettig maar niet noodzakelijk werd gevonden, zal het nut ervan veel beter aantoonbaar zijn.

Technische benadering

De in de strategiefase gekozen algemene oplossingsrichting zal na het instellen van de centrale data-administratie nader gedetailleerd kunnen worden door uitgaande van de centrale datadefinitie per systeem of per database (afhankelijk van onder andere de koppelingsgraad) de voorkeursoplossingen (meervoud) te bepalen. Deze detaillering zal echter niet al te ver doorgevoerd moeten worden om ruimte te laten voor de afweging tussen globaal optimale (uniforme) oplossingen versus lokaal optimale oplossingen.

Randvoorwaarden

Nadat de projectenportefeuille en de prioriteitstelling daarin zijn bepaald, dienen ook de randvoorwaarden als de te hanteren richtlijnen, procedures zoals voor versiebeheer, kwaliteitsbewaking en review-procedures te worden bepaald. Voorzover normaliter al organisatiebrede richtlijnen golden met per project de ruimte voor aanpassingen, zal voor het geheel van Jaar 2000-deelprojecten die ruimte wat minder zijn, omdat de projecten immers op de gemeenschappelijke doelen gericht moeten blijven.

Reclame maken

Het moge vreemd klinken voor een alles-of-niets situatie als het Jaar 2000-probleem, het zal hard nodig zijn om luid en duidelijk reclame te maken voor de Jaar 2000-projecten. Het is namelijk nodig om:

- steun van het management te krijgen in de vorm van voldoende budgetten en aandacht. Gezien de vereiste budgetten en de rentabiliteit van de projecten zal behoorlijk wat overredingskracht nodig zijn;
- begrip van de gebruikers te krijgen voor de door capaciteitsgebrek mogelijk verminderde service;
- steun (motivatie) te geven aan de ontwikke-

| Datum | Gebeurtenis |
|-------------------|---|
| Voorgaande jaren | Programma's beginnen mankementen te vertonen doordat data die over de eeuwgrens liggen, verkeerd worden verwerkt. |
| 1 januari 1999 | Programma's waarin het jaartal 99 duidt op 'eeuwigdurend', 'niet van toepassing' of dergelijke, beginnen fouten te produceren. |
| 9 september 1999 | Idem, inzake 9/9/99 |
| 1 januari 2000 | Het uur U, voor dan actieve systemen. |
| 3 januari 2000 ? | Eerste werkdag van het millennium?, de eerste dagelijks draaiende systemen worden gestart. |
| 29 februari 2000 | Schrikkelidag; weekdag- of maandaanduiding kan scheef gaan lopen. |
| Enig jaar na 2000 | Bijvoorbeeld: Computers die verkiezingen moeten verwerken, waren nog niet onderzocht (niet relevant – tijdgebrek) en gaan in de fout. |

laars van oplossingen voor hun enigszins monotone maar precisie vergende werk.

Tabel 3. Speciale data.

Analyse

Per project zal een nauwkeurige analyse moeten worden gemaakt van de erin aan te passen systemen. Onderstaande aandachtspunten dienen daarbij beslist te worden meegenomen.

Documentatie

Het verkrijgen van voldoende en up-to-date documentatie lijkt een vrij oninteressante routineklus. Als het goed is, valt die eenvoudig in een dossier te vinden. Met name voor oudere systemen zal de documentatie echter zelden op orde zijn en ook de kennis van die systemen zal wellicht niet meer in de organisatie aanwezig zijn. Hierdoor zal het nogal eens nodig zijn systeemdokumentatie eerst te vervaardigen met 'reverse engineering', voorwaar geen simpel en zeker geen klein karwei.

Teststelsysteem

Tezamen met het zo vroeg mogelijk verkrijgen van voldoende documentatie zal ook het bouwen van een teststelsysteem reeds in een zo vroeg mogelijke fase moeten worden gestart. Tot nu toe bleek het opzetten van een teststelsysteem namelijk in de praktijk meer werk te vragen dan de brochures van de leveranciers beloofden. Indien het ontwikkelen van het teststelsysteem pas wordt gestart na afronding van de programmeerwerkzaamheden, wordt kostbare tijd verdoen.

Test gevoeligheid

Een voor de hand liggend maar nogal eens vergeten onderwerp in de aanloopfase van Jaar 2000-projecten is het op voorhand testen van de gevoeligheid van een systeem voor 2000-overschrijdende jaartallen.

Enerzijds kan blijken dat de gevoeligheid meevalt. Dat zou kunnen betekenen dat de verwachtingen ten aanzien van de benodigde inspanning kunnen worden verlaagd, maar voordat een te gunstig beeld wordt geschapen dient dan wel te worden gecontroleerd dat de enkele gevoeligheden die naar voren komen niet binnen het systeem een grote onderzoeks- en wijzigingsinspanning blijken te

vergen. Wellicht zijn de enkele optredende storingen maar het topje van de ijsberg.

Anderzijds kan blijken dat het onderhanden systeem op nog onvermoede plaatsen afwijkingen genereert. In dat geval is het zaak de verwachtingen ten aanzien van de benodigde inspanningen direct naar boven bij te stellen. De inspanning kan uiteindelijk mee blijken te vallen doordat bijvoorbeeld per optredende fout maar weinig broncode behoeft te worden gewijzigd. Dit zal echter niet op voorhand vaststaan en voor de budgetten geldt als altijd: 'better be safe than sorry'.

Interfaces

De interfaces in of van het onderhanden systeem naar andere systemen zullen moeten worden beschouwd als onderdeel van het systeem zelf. Met name voor het bepalen van de optimale oplossingsrichting zullen de aangeleverde en afgenomen dataformaten een rol kunnen spelen. Door de interfaces bij de ontwerpfase van de wijzigingen te betrekken kan voorkomen worden dat het later nodig blijkt om een interface naar de interface te moeten ontwikkelen.

Gemeenschappelijke gegevens

Het voornoemde punt van interfaces geldt evenzo als sprake is van gemeenschappelijk gebruik van gegevens door diverse systemen. Bovendien zal in dit geval echter al rekening moeten worden gehouden met de andere dan het onderhanden systeem.

In dat geval zal het namelijk zeer slecht kunnen uitpakken dat voor het onderhanden systeem een optimale oplossing wordt gekozen, waarvan de gevolgen doorwerken in de gemeenschappelijke databases. Andere systemen zullen zich dan moeten conformeren aan die – wellicht voor die systemen verre van optimale – oplossing of er zullen extra interfaces moeten worden ontwikkeld.

Analyse van programmatuur

De analyse van de programmatuur in engere zin bestaat uit:

- het analyseren van de datastromen in het systeem;
- het opstellen van diagrammen van:
 - de logische structuur,
 - de datastructuren,
 - de relaties tussen programmatuur- en gegevenselementen;
- het decomponeren van de diagrammen tot een zo dicht mogelijk bij de broncode gelegen niveau;
- het analyseren van de programmalogica;
- het standaardiseren van aangetroffen of benodigde datumsubroutines.

Dit laatste vormt reeds een begin van de wijzigingsfase maar zal voorafgaand aan de feitelijke wijzigingen moeten worden gedaan om te voorkomen dat tijdens de programmatuurwijzigingen telkens aanpassingen in (het steeds grotere aantal reeds veranderde onderdelen van) de software zullen moeten worden gemaakt vanwege voortschrijdend inzicht. Dit voortschrijdende inzicht zal namelijk, als de wijzigingen bottom-up door bijvoorbeeld programmeurs worden aangedragen,

gemakkelijk gekleurd kunnen zijn door de specifieke deelproblemen waar die programmeurs op dat moment mee bezig zijn zonder dat de resulterende bijgestelde oplossing voor het gehele systeem optimaal hoeft te zijn.

Wijzigingen

Na alle voorbereiding en analysewerk zullen de feitelijke aanpassingen moeten worden doorgevoerd. Daarbij zijn te onderscheiden het ontwerpen van de wijzigingen, het daadwerkelijk wijzigen en het (liefst automatisch kunnen) testen van de wijzigingen. Tevens zal de documentatie moeten worden bijgewerkt.

Ontwerpen van wijzigingen

Bij het ontwerpen van de wijzigingen spelen de volgende zaken een rol:

- Het gebruik van intelligente zoekmiddelen kan een versnelling betekenen voor het opsporen van datum bewerkingen. Ze kunnen de eenvoudiger te vinden datum bewerkingen opsporen en daarmee juist een deel van het demotiverende lopende bandwerk overnemen. Zoals hieronder bij de paragraaf over de 'Silver Bullet' wordt verduidelijkt, zullen overigens zoekmiddelen, hoe intelligent ook, niet meer kunnen zijn dan hulpmiddelen die de analist ondersteunen.

- Behalve bij berekeningen zullen met name datumgerelateerde vergelijkingen per geval moeten worden onderzocht om te bezien of en zo ja, welke wijzigingen nodig zijn. Bovendien zal, met van geval tot geval verschillende diepgang, gezocht moeten worden naar verwijzingen, afwijkend gebruik van datumvelden en dergelijke.

- In sommige gevallen zullen datumvelden worden gebruikt in sleutelvelden. Het is van het grootste belang dat die gevallen in de programmatuur en in de documentatie op een niet te missen wijze worden gemarkeerd. Tijdens de ontwikkeling van interfaces en tijdens de gegevensconversies zullen de resultaten voor deze velden extra goed moeten worden gecontroleerd.

- Voorzover die nog beschikbaar zijn, zullen telkens opnieuw de oorspronkelijke eisen (of andere vormen van functionaliteitdefinities) aan de programmatuur moeten worden teruggezocht om vast te stellen of de wijzigingen neutraal zijn voor wat betreft de functionaliteit.

- Voorzover dat nog niet het geval is, zal de code moeten worden gemodulariseerd om het onderzoeksproces beheersbaar te houden. Door eerst te modulariseren en vervolgens de software te testen kunnen tevens nog onontdekte afhankelijkheden worden opgespoord.

Codewijzigingen

Het feitelijk wijzigen van de broncode behoeft geen bijzondere problemen op te leveren. Het ligt voor de hand de werkzaamheden op te splitsen in de volgende onderdelen:

- het aanpassen van schermen en overzichten;

- het aanpassen van batch-jobs (Job Control Language-modules);
- het genereren van nieuwe dialogen;
- het aanpassen van database-queries;
- het aanpassen van de programmacode (de 'business logic').

Op die wijze worden de wijzigingen vanaf het doel (gewijzigde output- en input-formaten) terug herleid tot programmawijzigingen, en zullen er dus niet te veel, achteraf gezien onnodige of verkeerd gekozen, wijzigingen in de business logic worden doorgevoerd.

Automatisch testen

Vanzelfsprekend zullen de wijzigingen getest moeten worden. Doordat – als het goed is – alle wijzigingen betrekking hebben op een beperkte verzameling Jaar 2000-problemen, kunnen de tests relatief eenvoudig in een testdata-bibliotheek worden ondergebracht. Dit bevordert de mogelijkheden tot geautomatiseerd testen.

Documenteren

Uiteraard is het, zoals altijd, van groot belang de doorgevoerde wijzigingen goed te documenteren, zowel in de broncode als in de systeem- en projectdocumentatie.

Naast het bijwerken van documentatie zal ook getraind moeten worden in het gebruik van gewijzigde systemen. Enerzijds zullen systeembeheerders en operators extra alert moeten worden gemaakt op het detecteren van optredende fouten, anderzijds zullen ook gebruikers moeten worden getraind in het mogelijk anders invoeren van gegevens en het interpreteren van eventuele (resterende) ambigue jaartallen.

Testen

'Test early, test often' is als altijd het devies. Zeker omdat niemand in de organisatie de deadline kan wegschuiven en er dus in principe geen vertragingen door te lange ontwikkeliteraties mogelijk zijn. Overigens wordt verwezen naar een artikel elders in deze Compact, waarin de testmaterie wordt beschreven.

Migratie

Ook het migratiepad is telkens wat complexer dan het normaliter is door de vaak vele interfaces die telkens moeten worden aangepast. In het migratietraject moet rekening worden gehouden met de volgende punten:

- Het voorbereiden van de handmatige dataconversies zal meer gedetailleerd moeten plaatsvinden dan anders. Ambiguiteiten zullen door alle betrokkenen op eenzelfde – liefst de goede – manier worden opgelost. Tevens zullen meer dan het gebruikelijke aantal controles worden ingepland.
- Ten aanzien van de voorbereiding van de automatische dataconversies zal het goede verloop van de uit te voeren conversies in de gaten moeten wor-

den gehouden omdat de interrelaties met eventueel benodigde hardware-uitbreidingen (opslagcapaciteit) voor vervelende storingen kunnen zorgen.

- De hardware en/of de operating-systemen zullen mogelijk aanpassing behoeven vanwege fouten in de dataroutines daarin. In dat geval dient te worden gecontroleerd dat alle (andere) systemen die op die hardware c.q. operating-systemen draaien al of niet tegelijkertijd moeten worden aangepast. Zodra dat het geval is, neemt uiteraard de complexiteit van de migratie aanmerkelijk toe.

Daarnaast kan het ook zijn dat er niet zozeer datumfouten in de hardware of operating-systemen voorkomen, maar dat wel bijvoorbeeld de opslagcapaciteit tekort schiet om alle datumvelden van vier jaarcijfers te voorzien. Met name voor wat betreft schijfruimte kan dit probleem optreden. De planning van de hardware-uitbreidingen of operating-systeemupgrades zal dan, zeker indien ook andere systemen van de desbetreffende platformen gebruik maken, nauwkeurig moeten worden opgezet en nageleefd.

Het is onmogelijk een volledig geautomatiseerde oplossing voor alle problemen te bedenken.

- Ten slotte zullen zoals te doen gebruikelijk ook de finale systeem-, migratie- en acceptatietests moeten worden gepland. Naarmate de Jaar 2000-projecten vorderen, zal daarmee steeds meer ervaring worden opgedaan zodat leereffecten zijn te verwachten.

- De eerste stap bij de feitelijke migratie is het waar nodig aanpassen van de productieprocedures. Hoewel op het eerste gezicht geen invloed is te verwachten van de invoering van Jaar 2000-bestede systemen, zullen toch enkele relatief kleine wijzigingen nodig zijn. Zo zullen er bij expansie naar vier jaarcijfers bijvoorbeeld meer backup-tapes moeten worden gevuld, zijn er meer interfaces te beheren en wordt het versiebeheer van systemen meer dan normaal op de proef gesteld.

- Vervolgens zal dan het productiesysteem zelf moeten worden geïnstalleerd. Dit kan conform de gangbare procedures worden afgehandeld – mits de benodigde hardware-aanpassingen integraal worden meegenomen en de installatie van de interfaces nauwlettend wordt gecontroleerd.

- Daarna zal de data geconverteerd moeten worden, automatisch waar het kan maar anders handmatig en tijdrovend. Ook de controles op de gewijzigde gegevens zullen handmatig moeten worden gecontroleerd, vanwege juist bij jaartallen voorstelbare misinterpretaties.

- Ten slotte zullen de finale systeem- en acceptatietests moeten worden uitgevoerd en dient het systeem in productie te worden genomen.

Evaluatie

Uiteraard zullen de Jaar 2000-(deel)projecten zoals alle projecten moeten worden geëvalueerd om succes- en faalfactoren te kunnen bepalen. Juist bij de Jaar 2000-problematiek zijn leereffecten te behalen: optredende problemen en oplossingen daarvoor zijn hooguit gedeeltelijk afhankelijk van de specifieke systemen (hun functionaliteit), maar zijn veel-er echte Jaar 2000-gerelateerde problemen. Elke oplossing kan dan ook in principe worden hergebruikt.

Overigens kunnen natuurlijk ook de 'lessons learned' op het gebied van projectmanagement in het algemeen nuttig blijken voor toekomstige of andere projecten.

DE 'SILVER BULLET'

De omvang van de inspanningen trekt natuurlijk allerlei softwarebedrijfjes die claimen de 'Silver Bullet' te hebben gemaakt, het magische hulpmiddel (tool) dat volautomatisch al uw programma-tuur Jaar 2000-bestendig zou maken.

Dergelijke claims zijn onzin. Zoals P. de Jager, de Amerikaanse Jaar 2000-goeroe, in zijn stuk 'Biting the Silver Bullet' (<http://www.year2000.com/archive/bullet.html>) aangeeft, is het onmogelijk een volledig geautomatiseerde oplossing voor alle problemen te bedenken. Sterker nog, geautomatiseerde hulpmiddelen kunnen slechts gedeeltelijk het werk uit handen nemen (zie tabel 4). Zijn voor-naamste argumenten zijn:

- Het is onbestaanbaar dat alle scherm- en print-layouts geautomatiseerd kunnen worden gewijzigd. Wat als een regel met een jaartalveld 'vol' is? En als gekozen is om te blijven werken met tweecijferige jaartallen blijft het probleem van ambiguïteit bestaan.
- Bovendien zullen gebruikers moeten worden geïnstrueerd om vier cijfers in te voeren of om de juiste interpretaties te kunnen maken. De gedachte dat deze training geheel geautomatiseerd zou kunnen worden uitgevoerd, is naïef.

Tabel 4. Besparingen door geautomatiseerde hulpmiddelen.

| Activiteit | Inspanning | Besparing | Effect |
|---------------|------------|-----------|--------|
| Zoeken | 5% | 80% | 4% |
| Analyse | 25% | 10% | 2,5% |
| Conversie | 30% | 20% | 6% |
| Testen | 35% | 40% | 14% |
| Implementatie | 5% | 20% | 1% |
| Totaal | 100% | | 27,5% |
| Handmatig | | | 72,5% |

- Een tool kan nooit een hardware datumprobleem oplossen.

- De conversie van oude data kan nooit volautomatisch gebeuren omdat er ambiguïteit in de data verscholen zit. Welke datum is 01/02/02? De instructie in welke context dat soort gegevens geïnterpreteerd moet worden, zal handmatig per variabele moeten worden aangegeven. Daarnaast: welk jaar is '25? Is dat 1925, een geboortedatum, of 2025, de einddatum van een obligatie? Ook deze ambiguïteit zal overal expliciet handmatig moeten worden aangegeven.

- Ook geautomatiseerde expansie naar vier cijfers in databases kan onmogelijk volautomatisch. De ambiguïteiten zoals hiervoor aangegeven, zullen ook in de databases handmatig van een interpretatie moeten worden voorzien. Een Silver Bullet zal ook nooit een oplossing kunnen geven voor de vereiste inschakeling van de extra opslagruimte die nodig zal zijn. En hardware data zijn niet te expanderen.

- Geen enkel tool kan *alle* programmeertalen en -dialecten begrijpen. Zoals al was opgemerkt: juist de kritieke systemen zullen nogal eens in een obscuur dialect zijn geschreven. Er zijn genoeg talen en dialecten waarmee überhaupt weinig mensen bekend zijn, laat staan tools.

- Software is voor mensen vaak al onbegrijpelijk, zeker bij gebrek aan documentatie, maar een tool begrijpt nog veel minder dan mensen. De speciale berekeningen van tabel 1 bijvoorbeeld zijn constructies die vrij verbreid zijn, een tool zal ze daarom wel herkennen als probleempunt. Er zijn echter nog veel meer vreemde constructies die met data te maken hebben die wel werken maar waarvan de logica nauwelijks meer te doorgronden is. En als het tool al zodanig kan worden geïnstrueerd dat het de diverse constructies in de organisatie aankant, zal alsnog ook een aantal testgevallen erop moeten worden toegespitst.

- Wat te doen met code die al jaren fouten bevatte die nog nooit zijn opgemerkt? Het is zeker niet uit te sluiten dat door 'verbeteringen' van het tool zulke fouten plotseling naar voren komen. Een tool zal dat soort fouten nooit kunnen opsporen, laat staan verbeteren.

- Sommige code bevat impliciete aannames omtrent de formaten van invoer- en uitvoerparameters. Neem bijvoorbeeld nieuwe-variabele = substring (oude-variabele, x, y). Duidt x op het begin van een jaarveld? Welke waarde bevat y voor de lengte van het te extraheren jaarveld? Verandert het formaat van oude-variabele door een aanpassing? Dit soort zaken kan weliswaar per geval in een tool worden ingebouwd, maar uittuttend zal dat per definitie nooit kunnen zijn.

- Sommige variabelen zullen nu eens wel, dan weer geen datum informatie bevatten. Dat soort variabele-hergebruik verdient geen schoonheidsprijs, maar komt op grote schaal voor. Een tool kan onmogelijk herkennen waar al of geen wijzigingen nodig en/of mogelijk zijn.

- Een tool kan nooit de aanpassingen aan die nodig zijn om speciale vlaggen als '99' of '00' goed te herprogrammeren. Wat als de vlag in een ander programma wordt gezet of gelezen?
- Een geautomatiseerd tool kan nooit de prioriteiten bepalen waarmee systemen zullen worden aangepakt.
- Een tool kan onmogelijk de aanpassingen van gegevensuitwisselingen met andere organisaties aan.
- Tests zullen nooit honderd procent kunnen worden geautomatiseerd.

Kortom, geautomatiseerde hulpmiddelen kunnen flink wat helpen, maar een Silver Bullet die alle problemen oplost kan niet bestaan. De softwareleverancier die dat beweert, heeft de problemen kennelijk niet goed begrepen en verkoopt al gauw een kat in de zak.

SAMENVATTING

Samenvattend kan worden gesteld dat de Jaar 2000-fouten vooral van technische aard zijn, maar in principe in alle elementen van alle informatiesystemen kunnen voorkomen. Behalve inzicht in en beheersing van de techniek zal daarom ook inzicht en beheersing nodig zijn van de organisatorische aspecten. Door een aantal specifieke Jaar 2000-problemen en de omvang van het werk zal alleen een nauwgezette sturing van de projecten op alle niveaus tot een bevredigend resultaat kunnen leiden. Daarbij valt niet te vertrouwen op een Silver Bullet waarmee alle technische problemen volautomatisch zouden kunnen worden opgelost.

Indien dan behalve de interne relaties tussen systemen ook de relaties van de systemen met leveranciers en afnemers worden aangepast en alle systemen goed worden getest, is alles gedaan om de millenniumwisseling met een wat geruster hart tegemoet te zien.

*Ir.dr.s. J. van der Vlugt
Is als EDP-auditor werkzaam
bij KPMG EDP Auditors.
Zijn aandachtsgebied ligt bij
advies voor en audit van
informatiebeveiligingsbeleid
en de planning en beheersing
van automatiseringsprojecten.
Daarnaast heeft hij zich
binnen de business unit
Technical Auditing gespecialiseerd
in de beveiliging en audit van
Windows NT-systemen.*

Testen en de Millennium-conversie

Drs. T. Koomen en drs. E. Broekman CISA

Testen is nogal eens een onderbelichte projectactiviteit. Ten onrechte, testen kan immers een belangrijke bijdrage leveren aan het verminderen van projectrisico's. Op de pijlers 'Fasering', 'Technieken', 'Infrastructuur' en 'Organisatie' wordt een testaanpak uiteengezet die bijdraagt aan het zeker stellen van de Millennium-bestendigheid van systemen.

INLEIDING

De aanstaande eeuwwisseling is voor veel bedrijven bedreigend, wegens de problemen die hun informatiesystemen daarmee kunnen hebben. Dit komt doordat het jaarveld in datums vaak maar twee posities bevat, zodat het informatiesysteem geen onderscheid kan maken tussen jaartallen tot 2000 en daarna, bijvoorbeeld '01' kan '2001' zijn of '1901'. In de media wordt veel aandacht besteed aan dit probleem en veel organisaties propageren hun 'eigen' oplossing (in dit artikel Millennium-conversie genoemd).

Hoewel de meeste organisaties die met het Millennium-probleem bezig zijn het belang van testen onderkennen (schattingen lopen uiteen van 30% tot 60% van de totale inspanning), is in hun totale aanpak het testen vaak onderbelicht. Als voorbeeld worden enkele symptomen van deze onderbelichting geschetst.

– *De relatie tussen de Millennium-aanpassingen en de integrale informatievoorziening en dus de bedrijfsvoering wordt onderschat.*

Het bedrijfsproces is aan verandering onderhevig en de informatiesystemen die dit proces ondersteunen moeten daarom mee veranderen. Echter, door slechte onderhoudbaarheid en het vaak afwezig zijn van (goede) documentatie is het aanpassen van de informatiesystemen vaak een moeilijke zaak (legacy-problematiek). Als gevolg van de Millennium-overgang moeten veel informatiesystemen in meerdere of mindere mate extra aangepast worden, dus parallel aan het 'going-concern' van regulier onderhoud en nieuwbouw. Door de schaalgrootte van het probleem en het onvoldoende inzichtelijk zijn van alle relaties tussen programma's, bestanden, systemen, etc. is hier een grote kans op het ontstaan van fouten. In combinatie met het feit dat de systemen vaak zeer belangrijk zijn voor de bedrijfsvoering, vormen de aanpassingen daarom een aanzienlijk risico.

Testen is nodig om te controleren of de systemen na de aanpassingen nog correct werken, met name ook in samenhang met de andere systemen. Eerste aandachtspunt hierbij is of de test überhaupt uitgevoerd kan worden, omdat de sterke mate van complexiteit van de omgeving en het onvoldoende inzicht kunnen leiden tot een niet-testbaar resultaat. Vervolgens is goed configuratiebeheer noodzakelijk voor de bewaking van de status van de verschillende systemen (is een systeem wel/niet Millennium-proof?).

– *Testen is een activiteit tijdens of ná aanpassen.*

Testen is een proces van plannen, voorbereiden, specificeren en uitvoeren, waarbij het testen al aan het begin van het traject geïntegreerd met de overige activiteiten dient te worden gezien.

– *De testaanpak gaat voornamelijk in op bepaalde technieken.*

Een testaanpak is meer dan enkel het hanteren van bepaalde technieken. Er dient een projectmatige aanpak (fasering) gehanteerd te worden, met aandacht voor organisatie en infrastructuur.

– De nadruk wordt sterk gelegd op de applicatiesoftware.

Een misvatting is dat de programmatuur het enige is wat aangepast moet worden. De datumproblematiek speelt ook in alle andere onderdelen van een informatiesysteem: gegevens, systeemsoftware, JCL, randapparatuur, outputformulieren, pakketten, end-user programmatuur, testware, etc. Dit heeft allerlei gevolgen, bijvoorbeeld conversie van gegevens, niet-Millennium-proof zijn van de systeemsoftware, verkeerde sorteringen in JCL's, randapparatuur die invoer weigert, etc. Testen mag zich dus niet beperken tot enkel de aangepaste programmatuur.

– Er bestaat onduidelijkheid over juistheid en volledigheid van de impactanalyse.

Diverse organisaties zijn al langere tijd bezig met het inventariseren van wat wel of juist niet aangepast moet worden. Het blijkt erg moeilijk om een consistent beeld te krijgen van de ernst en omvang van het probleem. Testen is een maatregel om bijvoorbeeld te constateren dat, wanneer de impactanalyse uitwijst dat een systeem niet aangepast hoeft te worden, dit systeem inderdaad Millennium-proof is.

– Er is een ongefundeerd vertrouwen in tools.

In de verschillende stadia van een Millennium-conversie wordt gebruik gemaakt van tools, zoals voor het bepalen van de te maken aanpassingen, het maken van de aanpassingen, het testen, etc. Er mag niet op vertrouwd worden dat deze tools allerlei tests overbodig maken omdat het automatisch gebeurt. Eerst dient vastgesteld (= getest!) te worden hoe volledig en juist deze tools zijn. Pas later kan dit bij bewezen werking achterwege blijven.

In al deze gevallen worden risico's gelopen voor de bedrijfsvoering, waarbij niet alleen gedacht moet worden aan niet-correct functionerende systemen, maar ook aan bijvoorbeeld onvoldoende performance (mogelijk als gevolg van een programma dat te traag is geworden omdat datums moeten worden geïnterpreteerd). Testen kan deze risico's verkleinen en biedt als zodanig meer zekerheid voor het primaire bedrijfsproces.

Om een dergelijk complex en omvangrijk testproces beheersbaar te houden is in de visie van IP/Software Control Testen een gestructureerde testaanpak een cruciale succesfactor. Hierbij wordt al aan het begin van het traject het testen geïntegreerd met de overige activiteiten gezien. De keuzen die in de eerste fasen van de Millennium-conversie worden gemaakt, hebben namelijk allerlei gevolgen voor de beheersbaarheid van de risico's. Belangrijk is dat een maatregel als testen zo efficiënt mogelijk wordt toegepast. Dit betekent dat reeds in de beginfase van het project testexpertise nodig is om in te schatten wat de gevolgen van een bepaalde conversiekeuze zijn voor het testen en te adviseren over alternatieven. Conversiekeuze A kan misschien wel een iets makkelijker bouwtraject tot gevolg hebben, maar een veel moeilijker testtraject dan keuze B. Andersom geredeneerd: de wijze waarop getest moet worden, kan gevolgen hebben voor het totale conversietraject.

In dit artikel wordt de visie en aanpak van IP/Soft-

ware Control Testen beschreven voor het testen van de Millennium-conversie. De aanpak is gebaseerd op TMap®, de TestManagement approach van IP/Software Control Testen voor het gestructureerd testen van informatiesystemen.

VISIE OP MILLENNIUM EN TESTEN

In deze paragraaf komen de fasen van de Millennium-conversie en de noodzaak van een daaraan gerelateerde gestructureerde testaanpak aan de orde.

Globale aanpak van de Millennium-conversie

Diverse organisaties hebben een aanpak om de Millennium-problematiek te lijf te gaan. Hoewel al deze oplossingen verschillend zijn, is wel een algemene lijn te herkennen. In het algemeen is de Millennium-aanpak als volgt:

Overall-inventarisatie en impactanalyse

In eerste instantie worden alle mogelijke betrokken informatiesystemen waarvoor de kans bestaat dat ze eeuwwisselingsproblemen hebben, geïdentificeerd en geïnventariseerd. Vervolgens wordt in een impactanalyse bepaald wat de consequenties zijn van de eeuwwisseling voor de systemen en worden de systemen over zogenaamde clusters verdeeld. De indeling kan plaatsvinden op grond van diverse criteria, zoals minimalisatie van interfaces, verantwoordelijkheden, risico's voor de bedrijfsprocessen, etc. en is bedoeld om het proces beheersbaar te kunnen maken. Het is in de meeste gevallen namelijk onmogelijk in één keer alle systemen voor Millennium te converteren, dus een gefaseerde aanpak is nodig.

Overall-strategiebepaling

In deze fase wordt een Plan van Aanpak opgesteld voor het vervolgtraject, met aandacht voor zaken als aanpassingsstrategie, testaanpak, volgorde en planning.

Per cluster

Detail-impactanalyse en Planning

Een gedetailleerde analyse wordt uitgevoerd om een volledig inzicht te krijgen in wat aangepast moet worden. Vervolgens wordt bepaald hoe dit aangepast moet worden en wordt voor de rest van het traject een planning opgesteld.

Conversie

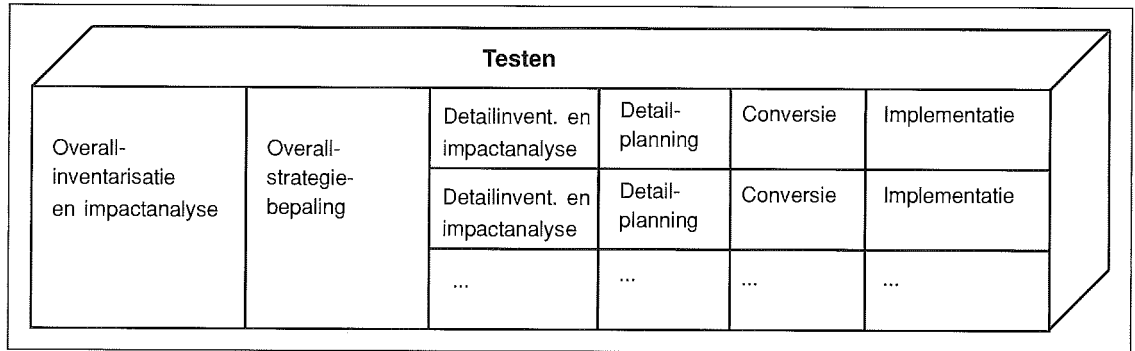
Op grond van de detailanalyse worden de aanpassingen geëffectueerd en getest.

Implementatie

Na acceptatie worden telkens één of meer informatiesyste(e)m(en) in productie genomen, met indien noodzakelijk conversie van gegevens, inpassing van tijdelijke oplossingen voor interfaces ('bridges'), etc.

Maar al te vaak ontbreekt in deze Millennium-aan-

Figuur 1. De fasen van de Millennium-conversie.



pak het aspect testen. Nu is het van groot belang om te onderkennen dat het testen niet één enkele activiteit is die ergens aan het eind van het traject plaatsvindt. Het testen is een continue activiteit die in *alle* hierboven genoemde fasen van de Millennium-conversie een rol speelt.

Specifieke Millennium-problematiek

De Millennium-conversie is in wezen gewoon een groot automatiseringsproject met alle gevaren en risico's die daarbij horen. Toch zijn er diverse kenmerken die een problematiek vormen die specifiek is voor de Millennium-conversie. Voor een groot deel geldt dit overigens ook voor andere bedrijfsbrede conversies, zoals bijvoorbeeld de conversie naar de Euro, de Europese munteenheid. Specifieke kenmerken van de Millennium-problematiek zijn:

Datums

De Millennium-problematiek is in essentie een datumprobleem. Datums zijn te vinden in vrijwel alle onderdelen van de systemen en worden voor vele doeleinden gebruikt. Systemen bevatten ook veel verborgen datums, bijvoorbeeld in coderingen. Dit betekent dat door alle systemen heen wijzigingen worden aangebracht.

Diversiteit/complexiteit (schaalgrootte)

Bij de objecten die aangepast moeten worden, moet niet enkel gedacht worden aan de applicatiesoftware, maar ook aan de systeemsoftware, JCL's, procedures, pakketten, end-user programmatuur, hardware, te converteren bestanden, tijdelijke overbruggingen (bridges), etc. De programma's en systemen hebben bovendien allerlei interfaces, die ook beïnvloed worden door de aan te brengen wijzigingen. De omvang van de aanpassingen en de volgorde waarin deze worden gemaakt, hebben allerlei gevolgen voor het inrichten van het testen.

Ongewijzigde functionaliteit

Een groot voordeel van de Millennium-aanpassingen is dat de oorspronkelijke functionaliteit ongewijzigd blijft. Dit vergemakkelijkt in aanzienlijke mate het testen omdat het oude systeem als referentiekader kan worden genomen. Sterk aangeraden wordt om dit voordeel niet ongedaan te maken door de Millennium-aanpassingen te combineren met regulier onderhoud, waarbij wél de functionaliteit gewijzigd wordt.

Harde einddatum

Omdat het jaar 2000 niet verschoven kan worden, heeft het project een harde deadline, waardoor er geen uitlooptmogelijkheden zijn. De deadline is overigens niet per definitie voor ieder systeem precies op 1 januari 2000. Veel problemen zullen al vóór 1/1/00 (1/1/2000 na conversie) gaan optreden, in andere gevallen ligt de deadline daarna.

Beschikbaarheid resources

Omdat naast de Millennium-projecten ook het reguliere onderhoud en nieuwbouw blijven plaatsvinden, is capaciteitsgebrek (in middelen en mensen) een voortdurende bedreiging.

Dit betekent vermoedelijk dat tot en met de eeuwwisseling voortdurend nieuwe mensen op de projecten ingezet worden, zodat het op peil houden van ervaring en knowhow een zorg is. Een nog groter risico is dat minder geschoold personeel op de Millennium-projecten wordt ingezet ('wat simpele testjes omdat de datums nu acht in plaats van zes posities zijn').

Tevens vormt de beschikbaarheid van testomgevingen een knelpunt, omdat een veelheid aan informatiesystemen en de relaties tussen deze systemen voor Millennium getest moet worden, en het reguliere onderhoud en nieuwbouw niet zullen stoppen.

Beheer/bewaking van Millennium-proof informatiesystemen

Het oplossen van de Millennium-problematiek betekent meestal niet dat het reguliere onderhoud en nieuwbouw tijdelijk worden stilgelegd. Dit heeft als gevolg dat regelmatig nieuw gebouwde of onderhouden systemen in productie gaan, nieuwe/aangepaste pakketten of systeemsoftware worden geïnstalleerd, etc. In al deze gevallen moet ook worden vastgesteld dat de aanpassingen *Millennium-proof* zijn.

Uitgangsdokumentatie voor de test lastig te identificeren

Aangezien vooral de oudere (Cobol-)programmatuur voor de eeuwwisseling aangepast moet worden, zal men vaak te maken krijgen met verouderde of niet meer aanwezige documentatie. Het 'normale' testen is gebaseerd op deze documentatie (werkt het programma volgens de specificaties).

Grote kans op regressiefouten

De wijzigingen moeten op veel plaatsen worden aangebracht, in programmatuur waarvan de documentatie vaak niet bruikbaar meer is en waarbij de programma's/systemen op allerlei manieren met

elkaar samenhangen. Hierdoor ontstaat een grote kans op regressiefouten.

Dit geheel maakt van de Millennium-conversie een complex en omvangrijk project. Om dit beheersbaar te houden, lijkt het welhaast noodzakelijk dat de Millennium-aanpassingen niet gecombineerd worden met andere vormen van onderhoud.

Gestructureerde testaanpak noodzakelijk

De Millennium-conversie is een complex en omvangrijk karwei met hoge bedrijfsrisico's. Het managen en beheren van het testproces hiervan is geen sinecure; de organisatie vraagt adequate beheersinstrumenten. Er moet simpelweg gedurende het hele traject antwoord gegeven kunnen worden op de klassieke vragen: wat, wanneer, wie, hoe, waarmee, etc. Dit wordt uiteraard niet bereikt door 'zomaar voor de vuist weg' te testen. Het toepassen van een gestructureerde testaanpak is pure noodzaak.

Een gestructureerde testaanpak is gebouwd op vier pijlers: een aan de ontwikkelingscyclus gerelateerde fasering van de testactiviteiten, een goede organisatorische inpasbaarheid, de juiste infrastructuur en hulpmiddelen en bruikbare technieken ter uitvoering van de activiteiten.

Bij IP/Software Control Testen is de gestructureerde testaanpak TMap® (een afkorting van Test Management Approach) ontwikkeld, die in detail invulling geeft aan de vier pijlers. Deze testaanpak vormt de basis van de meer specifieke Millennium-testaanpak, die in de volgende paragraaf in meer detail toegelicht zal worden.

Baten van Millennium-testen

De Millennium-conversie heeft als belangrijkste doel dat alles ná de eeuwwisseling nog net zo werkt als daarvoor. Bij deze weinig aansprekende doelstelling wordt al snel de vraag gesteld: 'Krijg ik nog meer waar voor mijn geld?' In geval van het testen kunnen nog de volgende baten worden genoemd:

- *Standaarden, richtlijnen, templates, invoering gestructureerd testen*
In de beginfase van het Millennium-testproces worden standaarden, richtlijnen, etc. opgesteld. Met geen of kleine aanpassingen kunnen deze procedures als de standaard 'fabrieksmatige' werkwijze voor het testen in de organisatie worden gehanteerd.
- *Regressietestware*
De regressietestware die wordt gecreëerd, is ook bruikbaar voor regulier onderhoud.
- *Testtools*
Het gebruik van de testtools is niet specifiek voor de Millennium-tests, maar kan ook in reguliere testtrajecten worden gebruikt.

Dit alles is een logisch voortvloeisel van het feit dat de testaanpak expliciet een fase Afronding heeft gedefinieerd om het testproces op een goede manier af te ronden en de testware voor volgende re-

leases (onderhoud) over te dragen aan de beheerorganisatie.

DE MILLENNIUM-TESTAANPAK VOLGENS TMAP®

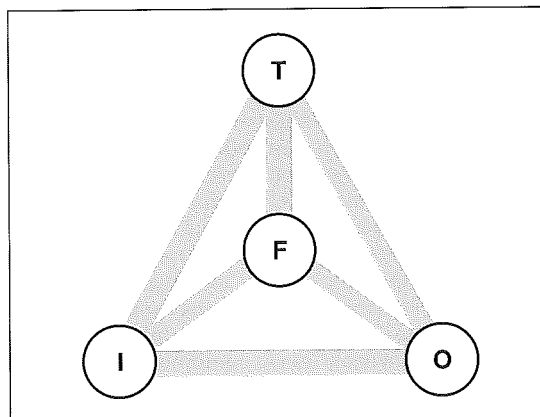
De hier beschreven gestructureerde testaanpak voor het Millennium is op basis van de vier pijlers van TMap® uitgewerkt. Bij elke pijler zijn maatregelen beschreven, die de al eerder geschetste specifieke Millennium-problemen aanpakken. De uitwerking richt zich dus primair tot het aandachtsgebied 'Millennium' en geeft geen volledig beeld van het organiseren van een gestructureerde testaanpak. Hiervoor verwijzen wij graag naar de standaardliteratuur over gestructureerd testen.

Bij Millennium-projecten zijn grote verschillen mogelijk in bijvoorbeeld omvang, organisatie, de relatie tussen de datum en het primaire bedrijfsproces, de beschikbare tijd/capaciteit en de 'testvolwassenheid' van de organisatie. Daarom moet de hier geschetste Millennium-testaanpak gezien worden als generieke aanpak, en zijn met name in de fase Inrichting testproces allerlei varianten mogelijk.

De pijler 'Fasering'

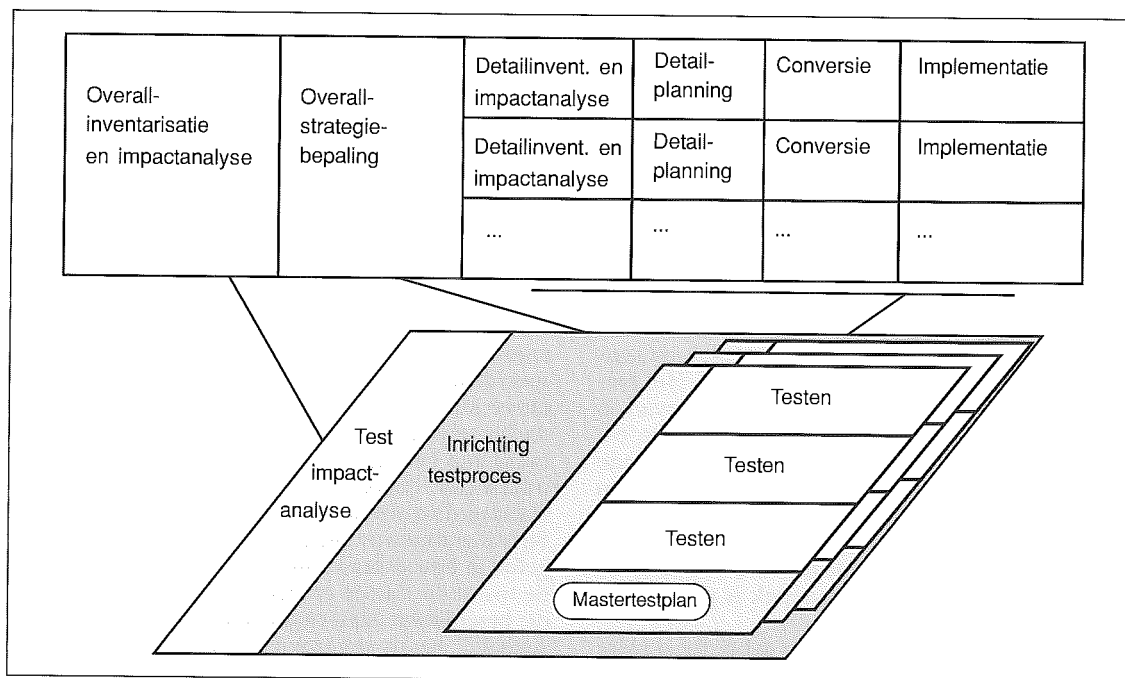
De faseringspijler beschrijft welke activiteiten wanneer uitgevoerd moeten worden. Deze pijler heeft een sterke koppeling met de andere drie pijlers: de fasering bevat ook activiteiten om te zorgen dat de andere drie pijlers voldoende ingericht zijn. Wanneer in deze paragraaf activiteiten worden beschreven die zijn voorzien van (I), (O) of (T), betekent dit dat verdere informatie is te vinden bij de beschrijving van de respectievelijke pijlers.

De testactiviteiten zijn dus vervat in een faseringsmodel, dat parallel aan de faseringsmodellen voor systeemontwikkeling te hanteren is. In het faseringsmodel zijn de drie hoofdactiviteiten van het testen (plannen, specificeren en uitvoeren) verdeeld over een vijftal fasen: Planning & Beheer, Voorbereiding, Specificatie, Uitvoering en Afronding. Voor de Millennium-aanpak worden deze fasen vooraf gegaan door twee extra fasen: Test impactanalyse en Inrichting testproces. Hiermee sluit de fasering van het testen goed aan op de gangbare globale aanpak van een Millennium-conversie.



Figuur 2. De vier pijlers onder een gestructureerde testaanpak.

Figuur 3. Testen in alle conversiefasen.



Ook wordt hiermee bereikt dat het testen een rol speelt in *alle* fasen van de Millennium-conversie. Figuur 3 illustreert dit.

Hierna wordt elk van de testfasen kort toegelicht, vooral gericht op aspecten die voor het Millennium-testen relevant zijn.

Test impactanalyse

Deze fase heeft tot doel om in een vroeg stadium van het project te adviseren over de inrichting van het totale conversieproces door te kijken naar de impact op testen. Typische aandachtspunten hierbij zijn:

- Worden alle testobjecten geïdentificeerd? Wordt de aandacht niet alleen op programma-tuur gericht maar ook op andere objecten als systeemsoftware, bestanden, etc.
- Moet de documentatie aangepast? Is deze nu al niet up-to-date, dan is aanpassen waarschijnlijk niet zinvol.
- Wordt bestaande testware in de inventarisatie meegenomen? (T)
- Kan voor een systeem/cluster wel een testomgeving gecreëerd worden? (I)
- Wordt de organisatorische inrichting zodanig dat effectief en efficiënt testen mogelijk is? (O)
- Welke tools worden bekeken, zijn deze ook voor testen bruikbaar?
- Is de testbaarheid niet dusdanig laag dat het vrijwel onmogelijk is te testen of het geconverteerde systeem kwalitatief goed werkt?
- Is een pilot voor testen voorzien?
- Wordt bij gegevensconversies rekening gehouden met performance/doorlooptijd?
- Worden Millennium-conversies gecombineerd met regulier onderhoud en/of met andere conversies (bijvoorbeeld naar ander platform)? (T) Indien de functionaliteit ongewijzigd blijft, vergemakkelijkt dit in aanzienlijke mate het testen. Daarnaast heeft combineren het grote nadeel dat de analyse van fouten erg complex kan worden.

Inrichting testproces

Wegens de omvang van de te converteren informatievoorziening wordt het testen van de Millennium-conversie meestal uitgevoerd in vele separate testprojecten. Het doel van de fase 'Inrichting testproces' is om al deze trajecten zo vlot en zo soepel mogelijk te laten verlopen. Zorg ervoor dat de verschillende trajecten elkaar niet in de weg zitten en laat ze zo veel mogelijk profiteren van de ervaringen en voorzieningen uit andere trajecten. Daartoe richt deze fase zich op de volgende drie gebieden:

- Zo veel mogelijk vooraf en eenmalig inrichten van het testproces, met als doel optimalisatie van de verschillende testprojecten binnen het totale Millennium-proces ('testfabriek').
- Bepalen prioriteiten en afhankelijkheden tussen de verschillende testprojecten in relatie tot de overall-strategiebepaling.
- Beheren en bewaken onderliggende Master-testplannen.

De inrichting van het testproces vindt parallel aan de overall-strategiebepaling voor de conversie plaats. Op grond van de informatie uit de Test impactanalyse worden de volgende onderwerpen behandeld:

- Opstellen en beheren van richtlijnen, standaarden, etc. (O)
 - Wat zijn de belangrijkste risico's en welke maatregelen zijn te nemen
- De risico's die gelopen worden bepalen aspecten als diepgang, aantal en volgorde van testen. Voor een applicatie die het primaire bedrijfsproces ondersteunt, wil men zo vroeg en zo veel mogelijk zekerheid dat de Millennium-problematiek correct is opgelost. In deze fase wordt het belang van de verschillende informatiesystemen geïnventariseerd. Het testen van de systemen wordt in een bepaalde volgorde uitgevoerd, waarbij vanuit testoogpunt gezien voorrang gegeven moet worden aan de belangrijkste systemen. Hierbij moeten alle test-

objecten (applicaties, systeemsoftware, gegevens, e.d.) beschouwd worden, aangezien bijvoorbeeld het testen van aangepaste programmatuur erg lastig (en kostbaar) kan zijn indien de onderliggende systeemsoftware nog niet Millennium-proof is. Ook kan het ene systeem al datumproblemen in 1997 vertonen, terwijl het andere systeem pas in 2000 of later problemen zal krijgen.

– Definiëren van de scope van de onderliggende Mastertestplannen

In de parallele fase Overall-strategiebepaling worden clusters van te converteren systemen opgesteld. In de volgende testfase worden de Mastertestplannen opgesteld. Er wordt niet per se een Mastertestplan opgesteld voor een geheel cluster. De scope van een Mastertestplan kan ook op één of meer systemen uit het cluster betrekking hebben. Overwegingen om voor een bepaalde groep systemen een separaat Mastertestplan te maken, zijn bijvoorbeeld:

- één verantwoordelijke persoon/afdeling voor het testen van een aantal systemen;
- gemeenschappelijke begin- en/of einddatum van testen;
- identieke benodigde infrastructuur.

– Generieke strategiebepaling (T)

In principe wordt per project een teststrategie bepaald. Om wildgroei in de gekozen oplossingen te voorkomen, worden binnen Millennium-testprojecten de keuzemogelijkheden beperkt door een generieke teststrategie op te stellen.

– Organisatorische mogelijkheden/verantwoordelijkheden (O)

– Beschikbaarheid testomgevingen (I)

Het uitvoeren van een pilot verdient sterke aanbeveling, om te toetsen of de gevolgde werkwijze correct is.

Mastertestplanning en Testen

Wanneer systemen de conversiefasen Detail-impactanalyse, Conversie en Implementatie ingaan, worden ook de Mastertestplannen opgesteld. In de fase Inrichting testproces is bepaald welke systemen gezamenlijk onder een Mastertestplan vallen. Voor iedere groep systemen zullen in principe verschillende soorten tests uitgevoerd worden door verschillende teams. Bijvoorbeeld: het ene team richt zich op de online functionaliteit, terwijl een ander team de koppeling met de batches op het mainframe onder handen neemt.

De bedoeling van een Mastertestplan is om de integratie, afstemming en coördinatie van al deze afzonderlijke tests te regelen. Daarbij is met name aandacht voor de volgende zaken:

- welke testsoorten¹ en -vormen² zijn relevant;
- het concreet uitwerken en uitbreiden van de generieke strategiebepaling;
- organisatie;
- infrastructuur;
- planning.

Vervolgens wordt iedere gedefinieerde afzonderlijke test daadwerkelijk uitgevoerd in de volgende fase Testen. Zij doorlopen het 'traditionele' traject

met de fasen Planning & Beheer, Voorbereiding, Specificatie, Uitvoering en Afronding.

De fasen Mastertestplanning en Testen zijn belangrijk, veelomvattend en omvangrijk, maar zijn voor het Millennium-testproject niet wezenlijk verschillend van de standaardaanpak. Voor meer details verwijzen wij daarom naar de standaardliteratuur.

De pijler 'Technieken'

Technieken bieden het testpersoneel beproefde, universele werkwijzen en bieden het management en de auditors de mogelijkheid het testproces inhoudelijk te volgen en te evalueren. Er bestaan vele verschillende technieken voor de vele verschillen-

*Door het uitvoeren van een pilot kan worden getoetst
of de gevolgde werkwijze correct is.*

de testactiviteiten in samenhang met de fasering. In dit artikel worden twee technieken nader belicht, wegens hun specifieke relatie met de Millennium-problematiek: strategiebepaling en een testspecificatietechniek.

Strategiebepaling

In de fase Inrichting testproces wordt een generieke teststrategie opgesteld. In de fasen Mastertestplanning en Testen vindt op basis hiervan verdere detaillering plaats. Het doel van het opstellen van een teststrategie is: het zo *vroeg* mogelijk vinden van de *belangrijkste fouten* op een zo *goedkoop* mogelijke wijze. Er moet dus een onderbouwde keuze gemaakt worden aangaande de vraag, waar de beschikbare capaciteit het best aan besteed kan worden. Globaal gezien worden de volgende stappen uitgevoerd:

1. Bepaal het relatieve belang van de te testen systemen of systeemdelen.
2. Bepaal het relatieve belang van de kwaliteitsattributen (te testen aspecten) van de systemen.
3. Bepaal welke testtechnieken daartoe het best toegepast kunnen worden en in welke afzonderlijke testtrajecten dit moet worden uitgevoerd.

Voor de Millennium-conversie zijn voor deze drie punten enkele algemene aanwijzingen te geven:

1. De vraag hoe diep of zwaar getest moet worden, is afhankelijk van een aantal factoren:

– Belang van het systeem voor de bedrijfsvoering

De gevolgen van (Millennium-)fouten zijn voor het ene systeem veel ernstiger dan voor het andere systeem. Dit betekent dat de testinspanning hier rekening mee moet houden, omdat testen erop gericht dient te zijn om de risico's te beperken.

– Complexiteit en omvang van de aanpassingen
Indien het systeem ingrijpend is aangepast, geeft

1. Een testsoort is een groep van testactiviteiten die gezamenlijk worden georganiseerd en aangestuurd. Voorbeelden van testsoorten: programma-, integratie- en acceptatietest.
2. Een testvorm is een groep activiteiten met het oogmerk het informatiesysteem op een aantal samenhangende kwaliteitsattributen te controleren. Voorbeelden van testvormen: regressie-, stress- en interfaces-test.

dit groter risico op fouten dan wanneer slechts op enkele plaatsen een aanpassing is gedaan. In geval van gegevensconversie zal deze ook getest dienen te worden.

- Zwaarte/diepgang van de andere tests
Dit laatste kan als volgt verduidelijkt worden: stel dat de Programma- en de Systeemtest (PT/ST) enkel het aangepaste systeem hebben getest met testgevallen ≤ 1999 , dan zal de Acceptatietest veel zwaarder dienen te zijn dan wanneer PT/ST ook met testgevallen ≥ 2000 zouden hebben getest. Van belang is dus dat er inzicht is in de diepgang en werkwijze van de afzonderlijke tests. Zonder dit inzicht zal of mogelijk veel zwaarder getest worden dan eigenlijk nodig is, of zullen risico's worden gelopen omdat ervan uitgegaan wordt dat iets vermoedelijk al wel getest zal zijn/nog getest gaat worden.

Wat hierbij ook speelt is dat fouten vinden in een laat stadium van het testen (Acceptatietest) twee belangrijke nadelen heeft:

- Omdat in dergelijke tests meestal grotere delen van het systeem worden bekeken, is analyse van de fout steeds moeilijker/tijdrovender.
- Herstel van de fout is relatief veel duurder, omdat het hele voorgaande traject weer doorlopen moet worden.

2. Voor de Millennium-conversie zijn er twee kwaliteitsattributen die vrijwel altijd de belangrijkste zullen zijn:

- Functionaliteit

In principe blijft de functionaliteit ongewijzigd door de Millennium-conversie. Hooguit de presentatie van gegevens (met name de datum) zal zichtbaar veranderd zijn.

- Performance

De conversie mag geen nadelige gevolgen voor de performance hebben. Daardoor zouden bijvoorbeeld tijdskritieke batch-processen namelijk een ernstig probleem kunnen gaan vormen.

Ook moet niet vergeten worden, dat de conversie-activiteit zelf voldoende snel uitgevoerd moet kunnen worden. Conversies vinden vaak plaats in weekends of avonden. Met name het converteren van grote productiebestanden kan wel eens niet binnen een dergelijk tijdsbestek passen.

3. Een testtechniek die bij uitstek geschikt is voor de Millennium-conversie, is de 'fototechniek'. Deze is specifiek ontworpen door IP/Software Control Testen ten behoeve van het Millennium-testen. Deze techniek zal hierna verder worden toegelicht.

Specificatietechnieken

Door middel van testspecificatietechnieken worden op een eenduidige en reproduceerbare wijze testgevallen afgeleid uit de uitgangsdokumentatie (functionele specificaties). Er is een duidelijk verband tussen de beschikbare uitgangsdokumentatie en de toe te passen testtechniek: naarmate de dokumentatie formeler gespecificeerd is, kunnen formelere testtechnieken toegepast worden.

Van veel testtechnieken die algemeen van opzet zijn, kunnen Millennium-varianties worden gemaakt. Voorbeelden hiervan zijn:

- algoritmetest;
- Code Inspection;
- syntactische/semantische test.

Een complicerende factor bij het testen van de Millennium-conversie is echter, dat veel oude systemen getest moeten worden waarvan de uitgangsdokumentatie niet voldoet of niet beschikbaar is. Vaak is het enige beschikbare uitgangspunt de applicatie zelf, dus de nog niet aangepaste executable/source.

Om toch de beschikking te krijgen over testgevallen en testdata voor de Millennium-tests, moet men terugvallen op bijvoorbeeld:

- Het opstellen van logische testgevallen op basis van de source

Testgevallen kunnen worden afgeleid van de niet-aangepaste source, waarbij de voorspellingen kunnen worden verkregen door het uitvoeren van het niet-aangepaste programma. Naar keuze kan zowel volledige dekking als partiële testdekking worden verkregen. Om de source inzichtelijk te maken kunnen mogelijk reverse engineering-tools worden ingezet.

- Gebruik maken van reeds beschikbare testdata
Bij de inventarisatie van aanwezige informatiesystemen wordt gekeken naar applicatiesoftware, pakketten, systeemsoftware, etc. Van belang is dat ook naar bestaande testware wordt gekeken, omdat dit later mogelijk gebruikt kan worden bij het testen. Er moet overigens rekening mee worden gehouden dat de testware ook geconverteerd moet worden.

- Gebruik van productiebestanden

In het algemeen heeft het gebruik van productiebestanden tijdens het testen niet de voorkeur. Redenen hiervoor zijn dat de dekkingsgraad niet bekend is en dat geen uitvoerovoorspellingen bekend zijn. In het geval van de Millennium-test kan met name het tweede nadeel enigszins worden genuanceerd: omdat de functionaliteit ongewijzigd is kan de 'oude' productie-uitvoer worden beschouwd als voorspelling van de testuitvoer. Voor het verkrijgen van inzicht in de dekkingsgraad kunnen mogelijk coverage analysis tools gebruikt worden. Er dient analyse plaats te vinden of het een representatief bestand betreft, en of eventueel aanvullingen noodzakelijk zijn.

- Invoergeneratie

Testgevallen kunnen ook gemaakt worden op basis van mogelijke invoervariaties. Dit levert meestal een groot aantal testgevallen op. Voor- en nadelen zijn gelijk aan die voor productiebestanden. Diverse tools kunnen het genereren van invoer ondersteunen, er zijn specifieke input-generatoren verkrijgbaar, maar ook 'oneigenlijke' tools als spreadsheets kunnen gebruikt worden.

Daarnaast is de testtechniek 'fototechniek' ontworpen, die in al deze situaties toepasbaar is.

Fototechniek

Deze techniek is met name geschikt om te testen of de verwerkingsgang na een conversie van het systeem niet ongewenst veranderd is. Het is een basis-

techniek waarop vele variaties gemaakt kunnen worden. In deze subparagraaf worden de hoofdlijnen van de techniek geschetst en enkele mogelijke variaties beschreven.

Basistechniek

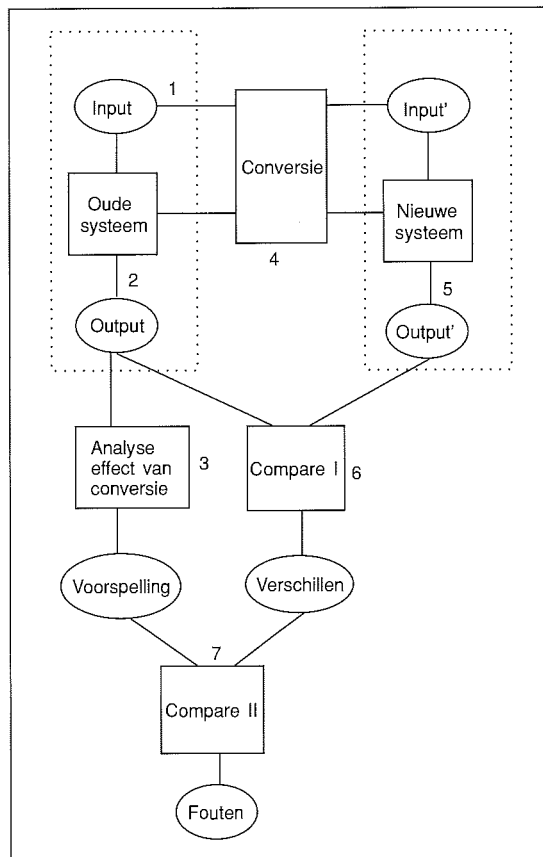
1. Stel een testset samen die door het oude systeem verwerkt moet worden. Zorg voor voldoende representativiteit in zowel de gegevens als de uit te voeren functies. Gebruik bij de keuze van datums niet uitsluitend bijzondere gevallen, zoals 31-12-1999. Zeer waarschijnlijk is voor het maken van de testgevallen, met name voor het 'doorschuiven' van de voorspellingen (zie Variaties), materiedeskundigheid onontbeerlijk.
2. Draai het oude systeem met deze testset als invoer en bewaar de uitvoer. Om de verwerking in volgende stappen exact identiek uit te kunnen voeren, is het aan te bevelen om in de voorbereiding deze verwerkingsgang vast te leggen met behulp van een record & playback-tool.
3. Analyseer wat het effect op de uitvoer zal zijn van de conversie van het systeem en van de testinvoer. Leid hieruit af, wat de verwachte veranderingen in de uitvoer van het geconverteerde systeem zullen zijn. Een voor de hand liggende verwachting is bijvoorbeeld, dat in veel schermen en lijsten de layout zal veranderen omdat in de gepresenteerde datums voortaan ook de eeuwaanduiding zichtbaar is.
4. Voer de conversie uit. Denk eraan, dat het niet alleen gaat om conversie van programmatuur, maar vooral ook van gegevens.
5. Laat de geconverteerde testinvoer verwerken door het geconverteerde systeem. Bewaar de uitvoer die hierdoor geproduceerd wordt, analoog aan stap 2.
6. Vergelijk de uitvoer van het geconverteerde systeem met die van het niet-geconverteerde systeem. In deze stap kunnen speciale tools ('comparators') heel nuttig zijn. Deze stap leidt tot een (waarschijnlijk groot) aantal geconstateerde verschillen.
7. Vergelijk de geconstateerde verschillen uit stap 6 met de voorspelde verschillen uit stap 3. De nu gevonden afwijkingen zijn de echte 'bevindingen' (in sommige organisaties heten ze 'fouten', 'problemen', 'issues', of iets dergelijks).

De genoemde stappen zijn in figuur 4 weergegeven.

In grote lijnen is dus het idee, dat er twee foto's gemaakt worden: één van het oude systeem en één van het nieuwe systeem. Deze foto's worden over elkaar gelegd, zodat de verschillen snel zichtbaar zijn en geanalyseerd kunnen worden. Op dit basis-idee zijn verschillende variaties mogelijk.

Variaties

- Er zijn vele variaties mogelijk in de testset die gebruikt wordt. De testgevallen kunnen afgeleid worden uit de source of kunnen speciaal ontworpen zijn (met bestaande technieken), of de bestaande testware kan toegepast worden.
- Er bestaat het risico dat de source-code van het



Figuur 4. De hoofdlijnen van de fototechniek.

systeem niet exact overeenkomt met de executable die in productie draait. Met name in legacy-systemen is soms wel eens wat 'gepatched'. Ook bestaat het risico, dat er na 2000 een nieuwe compiler gebruikt moet gaan worden, die de oorzaak van productieproblemen kan worden. Om dit te testen kan de volgende variatie worden uitgevoerd:

In plaats van direct het aangepaste systeem te draaien, wordt een tussenstap ingelast, waarin eerst het oude programma opnieuw gecompileerd wordt (eventueel met een nieuwe compiler). Deze executable moet vervolgens identiek werken aan het oude programma. Hiermee wordt vastgesteld dat de juiste combinatie van source-code, compiler en compiler-vlaggen is gebruikt. Pas wanneer dit zeker is, worden de Millennium-aanpassingen doorgevoerd.

- In grote lijnen zijn er twee verschillende soorten datums die afzonderlijk gemanipuleerd kunnen worden: de systeemdatum en de datums in (test)bestanden. Met deze datums zijn vele variaties in de test te maken. Enkele voorbeelden:

1. Zowel de systeemdatum als de transactiedatums in de testbestanden liggen vóór 2000. Met deze test wordt puur op regressie getest: 'Als we het geconverteerde systeem vóór het jaar 2000 al in productie nemen, is de werking dan nog identiek.'
2. De systeemdatum ligt na 2000 en de testbestanden bevatten datums van zowel voor als na het jaar 2000. Hiermee wordt pas getest of het systeem juist omgaat met datums in de volgende eeuw en dus Millennium-proof is. Om de uitvoer nog steeds te kunnen vergelij-

ken met de 'oude' uitvoer van het niet-geconverteerde systeem, worden *alle* datums in de testbestanden collectief met eenzelfde aantal jaren verhoogd. Hierbij kunnen speciale tools ('datumschuivers') zeer behulpzaam zijn. Wanneer doordeweekse dagen en zon- en feestdagen, etc. van invloed zijn op de functionaliteit, zorgt een verhoging met 28 jaar weer voor vergelijkbare resultaten. Immers, omdat alle condities en transacties in relatie tot de systeemdatum gelijk gebleven zijn, moeten de verwerkingsresultaten (niet de datums zelf, maar bijvoorbeeld wel de berekende bedragen) identiek zijn aan de 'oude' uitvoer.

- Optioneel kunnen de verschoven testdata ook gebruikt worden voor de niet-aangepaste programmatuur, om de werking na de Millennium-overgang te simuleren. Deze test dient dan als controle of aanvulling op de impactanalyse.
- De datums kunnen naar behoefte en mits de beschikbare capaciteit dit toestaat, verder gemanipuleerd worden om specifieke momenten te testen. Er valt bijvoorbeeld te denken aan:
 - 29-2-2000, omdat het jaar 2000 een schrikkeljaar is;
 - moment van eeuwwisseling.
- Net zoals de invoer (bestanden) geconverteerd kan worden, moet dat ook bij de uitvoer kunnen. Mogelijk is daar wel een speciaal conversie-tool of -programma voor nodig. De analyse van de resultaten van de conversie wordt nu eenvoudiger, omdat een rechtstreekse vergelijking gemaakt kan worden tussen de uitvoer van het geconverteerde (nieuwe) systeem en de geconverteerde uitvoer van het oude systeem. Dit lijkt wat op het principe van de 'vierkantstelling' bij boekhoudkundige controles. Men moet zich er echter wel van bewust zijn dat bij de conversie van de 'oude' uitvoer ook fouten geïntroduceerd kunnen worden.

De pijler 'Infrastructuur'

De infrastructuur voor het testen omvat alle faciliteiten en middelen die nodig zijn om naar behoren te kunnen testen. Hierbij kan onderscheid gemaakt worden in faciliteiten die nodig zijn om de test te kunnen uitvoeren (testomgeving), faciliteiten die nodig zijn ter ondersteuning van de testuitvoering (testtools) en de kantoorinrichting. Voor de testomgeving en de testtools zullen enkele Millennium-specifieke aandachtspunten beschreven worden.

Testomgeving

Systemen/applicaties/databases die draaien op een grote diversiteit aan systeemsoftware en hardware kunnen weliswaar wel geconverteerd worden, maar het inrichten van een testomgeving kan niet haalbaar blijken door de zware eisen die bepaalde tests (bijvoorbeeld productietests of tests op de samenhang tussen meerdere systemen) stellen. De tester dient dit zo vroeg mogelijk te signaleren. Indien dergelijke testomgevingen wel beschikbaar zijn, moet afstemming plaatsvinden zodat niet voor verschillende systemen een dergelijke test in dezelfde tijd gepland wordt.

Bij het definiëren van de benodigde testomgevingen zijn voor Millennium de volgende aspecten specifiek van belang:

- Het kunnen manipuleren van de systeemdatum stelt speciale eisen aan de testomgeving.
- Wanneer onderliggende systeemsoftware, DBMS, etc. nog niet Millennium-proof zijn, kan dit het testen van Millennium-aanpassingen in de applicaties erg moeilijk of zelfs onmogelijk maken.
- Om de aansluitingen met andere systemen te kunnen testen, moet hiermee rekening worden gehouden bij het inrichten van de testomgeving.
- Gezien de beperkte beschikbaarheid van grote testomgevingen moeten de verschillende deelprojecten goed het gebruik van dergelijke omgevingen op elkaar afstemmen. Hier zal in de fasen Inrichting testproces en Mastertestplanning rekening mee moeten worden gehouden.
- Bij gegevensconversies dient bij het opzetten van de omgeving rekening te worden gehouden met de performance/doorlooptijd.

Testtools

De grote omvang van Millennium-projecten in combinatie met ongewijzigde functionaliteit is een belangrijk argument voor het gebruik van testtools. Tools kunnen ondersteuning bieden aan vele verschillende testactiviteiten, zoals planning en beheer, het opbouwen van uitgangsbestanden, testuitvoering en beoordeling, etc. Hieronder staan enkele soorten tools genoemd die in het bijzonder geschikt zijn voor de Millennium-conversie.

- Reverse engineering-tools
Om op basis van de niet-aangepaste source testgevallen te maken, dient er inzicht te zijn in wat het programma doet, en hoe dit gedaan wordt. Hoewel RE-tools niet specifiek voor testen bedoeld zijn, kunnen dergelijke tools wel behulpzaam zijn bij Millennium-tests. De tools kunnen een programma inzichtelijker maken, bijvoorbeeld door het programma te tonen in de vorm van program-flow of Nassi-Schneidermann, of door de gebruiker in staat te stellen bepaalde gegevens makkelijk door het programma te volgen (data-trailing).
- 'Datumschuivers'
Bij de fototechniek dienen datums in testbestanden een aantal jaren verhoogd te worden. Tools kunnen dergelijke aanpassingen automatiseren. Overigens zijn ook niet-specifieke tools hiervoor heel bruikbaar, zoals Easytrieve, spreadsheets, etc.

- Systeemdatum-manipulators
Bij het Millennium-testen moet de systeemdatum gemanipuleerd worden, om het gedrag van de applicatie vóór en ná 2000 te controleren. Met name in de mainframe-omgevingen kan het erg lastig of zelfs onmogelijk zijn om de systeemklok aan te passen. Er zijn tools die opvragingen van de systeemklok onderscheppen, om op die manier een manipuleerbare datum aan het programma te kunnen teruggeven.

– Record & playback

Deze tools nemen als een soort recorder de testinvoer (gegevens en acties) op, zodat de test op een later tijdstip gemakkelijk kan worden herhaald. Over het algemeen worden record & playback-tools gecombineerd met comparators om het analyseren van de testresultaten mogelijk te maken. Een record & playback-tool is uitermate geschikt voor het regressietesten, welke test veel voorkomt in Millennium-projecten. Een groot voordeel is ook dat het tool in combinatie met de regressietestware ook ná 2000, of voor andere doeleinden dan Millennium, gebruikt kan worden.

– Comparators

Aangezien de functionaliteit niet of nauwelijks verandert, zal ook de testuitvoer van de geconverteerde systemen niet of nauwelijks veranderen. Comparators kunnen helpen om uit de grote hoeveelheden uitvoer de (hopelijk) kleine hoeveelheden verschillen op te sporen. Vaak zijn er verschillende comparators nodig voor de verschillende soorten uitvoer, zoals schermuitvoer, databases en files.

– Coverage Analyser

Omdat bij Millennium-tests vaak het oude programma als testbasis dient, en bestaande testware of productiebestanden als testgevallen gebruikt gaan worden, is het goed te weten welke dekking deze testgevallen nu hebben: wordt 25% of 85% van de code geraakt, en welk deel van de code wordt wel/niet geraakt. Op basis hiervan kan een risico-afweging plaatsvinden, waarna wellicht nog testgevallen worden toegevoegd.

De pijler 'Organisatie'

Een belangrijke voorwaarde voor een kwalitatief goed testproces is een goede organisatie. Bij de inrichting en de beheersing van de testorganisatie spelen diverse factoren een rol, zoals de betrokkenheid van veel verschillende disciplines, de tegenstrijdige belangen, de onvoorspelbaarheid, de ingewikkelde beheertaken, het gebrek aan ervaring(cijfers) en de tijdsdruk.

De organisatie van gestructureerd testen heeft een breed aandachtsgebied, waarvan wij de volgende onderdelen zullen belichten: het operationele testproces, de structurele testorganisatie, testbeheer en personeel & opleidingen.

Het operationele testproces

De vroege aandacht voor de organisatie in de eerste fasen verzekert dat voorbereidende activiteiten zo vroeg mogelijk starten, zodat de inzet van resources voor het testen zo efficiënt mogelijk gebeurt.

Er is ook capaciteit in de vorm van materiedeskundigen vereist, om te helpen met de testvoorspellingen ná het jaar 2000.

Over het schatten van de benodigde testinspanning is momenteel onvoldoende bekend. Huidige begrotingstechnieken (zoals FPA en TPA®) zijn niet toepasbaar omdat geen functionele aanpassingen worden gedaan. Mogelijke manieren om de inspanning te begroten zijn:

- percentage van de totale begrote inspanning

nemen. In de literatuur worden getallen van 35% of nog meer genoemd;

- resultaten van de testpilot extrapoleren;
- inschatten van de uren voor de afzonderlijke activiteiten en deze vervolgens extrapoleren;
- herleiden tot percentages per testsoort (programma-, integratie-, systeem- en acceptatietest);
- standaardverhoudingen bij testen hanteren (10% Voorbereiding, 40% Specificatie, 45% Uitvoering inclusief één hertest, 5% Afronding), de overhead op 10-15% begroten.

De structurele testorganisatie

Van belang bij de inrichting van het Millennium-conversietraject is dat organisatie en verantwoordelijkheden duidelijk zijn, met name op het aspect van kwaliteit. Wanneer een applicatie uit een conversiefabriek komt, en tijdens het testen worden problemen geconstateerd, dient duidelijk te zijn wanneer en door wie deze worden opgelost. Hierbij kan zowel worden gedacht aan de wijze van clustervorming als aan de wijze waarop de Millennium-conversie ingericht wordt: projectmatig, in regulier onderhoud of fabrieksmatig.

Testen in regulier onderhoud heeft als nadeel dat het mixen van de Millennium-aanpassingen met functionele aanpassingen het gebruik van de fototechniek veel lastiger maakt. Daarnaast is het moeilijker om een gestructureerde testaanpak te introduceren, met als gevolg dat het testproces minder goed beheersbaar is.

Met name in mainframe-omgevingen kan het erg lastig of zelfs onmogelijk zijn om de systeemklok aan te passen.

In de fase Inrichting testproces worden voor alle Millennium aan te passen informatiesystemen richtlijnen en technieken neergezet over de te hanteren testsoorten, strategie-overwegingen, planings- en beheersaspecten, infrastructuur, organisatie, tools, technieken, evaluaties, etc. Deze 'testfabriek'-aanpak voorkomt dat diverse afzonderlijke Millennium-projecten het wiel telkens opnieuw gaan uitvinden, met andere woorden, testtechnieken, beheersprocedures, tools, etc. worden eenmalig opgezet, en kunnen vervolgens centraal ondersteund worden. Dit betekent niet dat een eenmaal vastgelegde richtlijn of procedure niet meer veranderd kan worden, maar juist dat nieuwe inzichten vanuit afzonderlijke projecten hier bij elkaar komen, en weer verspreid kunnen worden naar andere projecten. Ook kan op dit niveau gedacht worden aan een testsupport-center, van waaruit bijvoorbeeld ondersteuning aan projecten gegeven kan worden.

Testbeheer

Het beheer van de externe producten (testobjecten en uitgangsdokumentatie) is geen verantwoordelijkheid van de tester. Het is echter zaak daar wél eisen aan te stellen (in de randvoorwaarden en uitgangspunten van de testplannen). Zo moeten bin-

Drs. T. Koomen

Heeft in de afgelopen tien jaar ruime ervaring opgedaan op vele terreinen van het automatiseringsgebied. Sinds vijf jaar heeft hij zich gespecialiseerd in het testen van informatiesystemen. Zowel binnen IP/ als daarbuiten geldt hij als expert op het gebied van de Millennium-problematiek.

Drs. E. Broekman CISA

Is meer dan tien jaar actief in het vakgebied Testen van informatiesystemen. Sinds zes jaar werkt hij bij IP/, waar hij opdrachten uitvoert variërend van test-consultancy tot het inrichten en managen van grote testprojecten. Hij was betrokken bij de ontwikkeling van TMap®, de de facto standaard in Nederland voor gestructureerd testen.

Op dit moment maken beide heren deel uit van het R&D-team dat verantwoordelijk is voor innovaties van TMap® met betrekking tot nieuwe technologieën en methoden zoals objectoriëntatie, RAD, computer/telefoon-integratie en de Millennium-conversie.

nenkomende producten identificeerbaar zijn (inclusief de versie) en moet het testproject invloed hebben op de prioriteitstelling bij het doorvoeren van wijzigingen. Door het grote aantal testobjecten (applicaties, systeemsoftware, bestanden, e.d.) en alle relaties tussen deze objecten is dit vermoedelijk één van de grootste problemen in de Millennium-problematiek. Men moet van elk testobject weten of het al Millennium-proof is, of er sinds de Millennium-aanpassingen nog regulier onderhoud op is uitgevoerd, en of het al getest is in combinatie met andere testobjecten.

Personeel & opleidingen

Het testpersoneel dient opgeleid te worden voor de in het Millennium-project geselecteerde testtechnieken en -tools. Ook kan gedacht worden aan een roulatiesysteem voor de beschikbare capaciteit en expertise op het gebied van testen, materiedeskundigheid en techniek.

SAMENVATTING

Het besef begint te groeien, dat de aanstaande eeuwwisseling voor veel informatiesystemen een probleem is en daarmee een bedreiging vormt voor vrijwel ieder bedrijf. De noodzakelijke conversie is een automatiseringsproject van uitzonderlijk grote omvang en complexiteit. Bovendien moet dit project naast de dagelijkse informatievoorziening (en het reguliere onderhoud daarop) plaatsvinden. Uit het oogpunt van beheersbaarheid en testbaarheid is het toch van belang dat de Millennium-aanpassingen niet gecombineerd worden met aanpassingen van andere vormen van onderhoud.

Om het testproces van een dergelijk automatiseringsproject beheersbaar te houden, is een gestructureerde testaanpak absolute noodzaak. Een solide basis hiervoor is een standaardaanpak zoals TMap®, die de vier pijlers van een gestructureerd testproces definieert: 'Fasering', 'Technieken', 'Infrastructuur' en 'Organisatie'. Om de Millennium-problematiek het hoofd te bieden zijn in dit artikel voor elk van de pijlers voorbeelden gegeven van Millennium-specifieke maatregelen.

Essentieel is, dat het testen vanaf het eerste begin actief betrokken wordt bij het conversietraject en men op dat punt niet een afwachende houding aanneemt.

LITERATUUR

[Aalb96] H. Aalbers, *Aanpak 2000*, IP/-seminar 'Jaar 2000, de millennium miljarden', 1996.

[Cap96] Cap Volmac TransMillennium Services, *Is uw bedrijf klaar voor het jaar 2000*, Euroforum-seminar 'Het jaar 2000: uw computersysteem van slag', 1996.

[Comp96] Computer Associates, *The CA Discovery 2000™ Solution*, Euroforum-seminar 'Het jaar 2000: uw computersysteem van slag', 1996.

[Comp96] Compuware, *Compuware Year 2000 Methodology*, Euroforum-seminar 'Het jaar 2000: uw computersysteem van slag', 1996.

[Grah96] D. Graham, P. Herzlich en C. Morelli, *CAST Report Third Edition*, CMI, 1996.

[Pol95] M. Pol, R. Teunissen, E. van Veenendaal, *Testen volgens TMap®*, Tutein Nolthenius, 's-Hertogenbosch 1995.

[Pol96] M. Pol, R. Teunissen, E. van Veenendaal, *Gestructureerd testen: een introductie tot TMap®*, Tutein Nolthenius, 's-Hertogenbosch 1996.

[Uden96] W. van Uden, Presentatie *Testen 2000*, Euroforum-seminar 'Het jaar 2000: uw computersysteem van slag', 1996.

[Valu96] Value Added Consultancy, *Micro Focus Challenge 2000*, Euroforum-seminar 'Het jaar 2000: uw computersysteem van slag', 1996.

[Your95] E. Yourdon, *Year-2000 Maintenance Iceberg*, Vol. VII, No. 6 of Application Development Strategies™, Cutter Information Corp., 1995.

'2000', een millennium om naar toe te werken

O. Venhuis

Een goede aanpak van de Jaar 2000-problemen kan niet zonder risico-analyses. Juist dergelijke impactanalyses kunnen de deelprojecten zó richten dat een gestructureerde projectaanpak mogelijk is.

INLEIDING

Verstoring in een informatiesysteem of het uitvalen van een informatiesysteem kan grote gevolgen hebben voor de betrouwbaarheid en continuïteit van de gegevensverwerking. Hierbij kan zelfs het voortbestaan van de onderneming in gevaar komen. Deze situatie kan zich voordoen bij de eeuwwisseling.

In vrijwel alle informatiesystemen wordt gebruik gemaakt van datumregistraties, welke onder andere worden gebruikt bij het uitvoeren van berekeningen en sorteringen. Deze datumvelden zijn vaak (nog) geprogrammeerd in zes posities, waarbij van het jaartal slechts de laatste twee posities zijn opgenomen, bijvoorbeeld in DDYYMM, MMDDYY of YYMMDD voor zogenaamde Gregorian dates en in YYDDD voor zogenaamde Julian dates (dat zijn dus andere begrippen dan Gregoriaanse of Juliaanse kalenderdata). Dit houdt in dat na 1999 (notatie 99) als jaarnotering 00 wordt vermeld. Door de geautomatiseerde systemen wordt in hun bewerkingen 00 gezien als een waarde die lager is dan 99, met alle gevolgen van dien.

De eeuwwisseling zal impact hebben op vrijwel alle bedrijfsprocessen, waarbij de mogelijke problemen te clusteren zijn in drie aandachtsgebieden, te weten:

- informatiesystemen: hardware, systeemsoftware, toepassingssoftware (standaardpakketten en maatwerkapplicaties), PC-applicaties, netwerken, randapparatuur en gegevens;
- infrastructuur: telefooncentrales en beheersystemen, audio/video-apparatuur en draadloze communicatiesystemen, waaronder mobiele communicatie en satellietcommunicatie;
- overige faciliteiten: niet-primaire bedrijfsvoorzieningen, waaronder stroomvoorzieningen, beveiliging, liften en klimaatbeheersing.

Recent verschijnen diverse berichten over het millenniumprobleem in de media. In deze artikelen wordt aangegeven dat een tijdige aanpak wordt vereist willen bedrijven niet in de moeilijkheden komen.

De Gartner Group voorspelde dat twintig procent van de computerapplicaties van het bedrijfsleven in 1996 problemen met data zou hebben. In 1999 is dit percentage opgelopen tot negentig. Deze problemen worden veroorzaakt doordat er nu al registraties plaatsvinden met een datum in of na het jaar 2000.

Onderzoek heeft uitgewezen dat in de Verenigde Staten momenteel slechts twintig procent van de bedrijven bezig is met het in kaart brengen van de problematiek. In Europa betreft het slechts tien procent van de bedrijven.

Slechts een beperkt aantal bedrijven onderneemt dus actie om de eeuwwisselingsproblematiek aan te pakken. De bedrijven die op dit moment nog geen actie ondernemen zijn te onderscheiden in een aantal groepen, namelijk bedrijven die de risico's die kunnen ontstaan (nog) niet onderkennen, bedrijven die de kennis (nog) niet hebben om de

problematiek aan te pakken en bedrijven die de problematiek wel onderkennen, maar ervan uitgaan nog voldoende tijd te hebben voor een aanpak van het probleem.

Er zullen bedrijven zijn die mogelijk geen problemen ondervinden van de eeuwwisseling. Het is echter ook voor deze bedrijven aan te bevelen om wel een analyse op de informatiesystemen uit te voeren om vast te stellen of dit inderdaad het geval is.

Door de bedrijven die nog niet tot actie zijn overgegaan, worden verschillende redenen aangevoerd. Hierbij wordt vaak onderschat dat problemen al vóór het jaar 2000 kunnen ontstaan. Er worden redenen aangevoerd als:

- Het systeem bestaat niet meer op 1 januari 2000. Dus waarom zouden we iets aanpassen wat geen problemen geeft?
- Het is niet mijn probleem. Ik zit bij de eeuwwisseling toch niet meer in deze functie. Waarom zou ik mijn kortetermijnbudget opofferen voor een langetermijnprobleem?
- Ik heb het te druk. We hebben onze handen vol aan onderhoudswerkzaamheden. Het millenniumprobleem kan ik er niet meer bij nemen.
- Ik wil het niet weten. Het probleem kon wel eens groter zijn dan we verwachten. Het management zal zich dan afvragen waarom we niet eerder zijn begonnen.
- Het valt wel mee. Alleen het datumveld hoeft maar te worden aangepast, dat kan toch niet zoveel werk zijn. Maak het probleem niet groter dan dat het is.
- Waarom zo'n haast? Over zes maanden of een jaar aanpassen is tijdig genoeg.
- We gebruiken databases van grote leveranciers, zoals DB2 en Oracle, dus hebben we het probleem niet.

Ook vanuit adviesorganisaties en automatiseringsfirma's bestaat op dit moment de nodige aandacht voor de problematiek. Door een aantal van deze instellingen worden, al dan niet in samenwerkingsverband, diensten aangeboden voor de aanpak van de problematiek.

De diensten hebben over het algemeen slechts betrekking op een beperkt aantal stappen in de probleemaanpak, waarbij onder andere met behulp van speciaal ontwikkelde software de informatiesystemen van de cliënt worden doorlopen op datumvelden.

Daarom is de vraag: 'Is het mogelijk te komen tot een min of meer uniforme aanpak op basis waarvan de millenniumproblematiek op beheersbare wijze kan worden opgelost?'

UITGANGSPUNTEN

Bij het gebruik van geautomatiseerde informatiesystemen is binnen een bedrijf een aantal functionele organisaties te onderscheiden, te weten de Gebruikersorganisatie (GO), de Verwerkings- en

transportorganisatie (VO) en de Systeemontwikkelingsorganisatie (SO). De GO is verantwoordelijk voor met name de betrouwbaarheid en de continuïteit van de gegevensverwerking. De GO heeft voor het uitvoeren van de gegevensverwerking de beschikking over diverse informatiesystemen, waarvan zij de operationele activiteiten, waaronder de verwerking, de opslag en het transport van de gegevens, heeft uitbesteed aan de VO. De VO heeft als rekencentrum de beschikking over diverse hulpmiddelen voor de uitvoering van haar werkzaamheden. Onder deze hulpmiddelen vallen de hardware en de systeemsoftware. In geval van ontwikkeling en onderhoud van software ligt de taak bij de SO.

De eeuwwisselingsproblematiek heeft op zowel de GO, de VO als de SO de nodige impact. De GO heeft het probleem dat zij niet weet of haar geautomatiseerde gegevensverwerking zal worden verstoord door de eeuwwisseling. De VO heeft deze onzekerheid ten aanzien van de hardware en de systeemsoftware. De VO en/of de SO zullen in opdracht van de GO de analyse op de informatiesystemen uitvoeren, waarna de aanpassing van de informatiesystemen zal gebeuren onder verantwoordelijkheid van de SO. Ook bij de aanpassing van systemen kan gebruik worden gemaakt van out-sourcing.

In de uitwerking van de methode van aanpak zal verder geen uitsplitsing meer worden gemaakt naar de verschillende organisatie-onderdelen. In iedere projectfase kan echter een toepasselijke taakverdeling worden gemaakt die organisatieproblemen voorkomt.

PLAN VAN AANPAK

In dit artikel is een methode van aanpak geschetst welke gebruikt kan worden om de eeuwwisselingsproblematiek op een beheersbare wijze op te lossen. De aanpak kan gelden als een min of meer normatieve aanpak.

Aangezien binnen Nederland op dit moment slechts een nog steeds onvoldoende aantal bedrijven is gestart met de aanpak van het probleem, is de normatieve aanpak met name geformuleerd vanuit de diverse in de media gepubliceerde artikelen, alsmede vanuit enkele seminars en praktijkervaring.

Allereerst wordt een schets gegeven van het ontstaan van de problematiek, alsmede een indruk van de mogelijke gevolgen en een inleiding tot het uitvoeren van de risico-analyse. Hierbij wordt ingegaan op de wijze waarop het datumprobleem kan worden opgelost. Daarna zal de aanpak worden uitgewerkt voor de beheersing van het millenniumprobleem.

Oorzaak en omvang

De eerste computers zijn in de jaren vijftig in gebruik genomen. Deze machines hadden een werk-

geheugen van slechts 1 tot 2 Kb. De computers van begin jaren zestig hadden in eerste instantie een werkgeheugen van niet meer dan 12 of 64 Kb. Het programmeren van deze computers vond plaats in machinetaal, hetgeen veel tijd in beslag nam. Daarnaast waren er weinig goede programmeurs beschikbaar. Hierdoor nam de noodzaak toe om de computers in een betere taal te programmeren. Eén van de talen die toen werd ontwikkeld, en die nog steeds in veel applicaties wordt gebruikt, is COBOL. Doordat banken en verzekeringsmaatschappijen (meer) begonnen te automatiseren, nam het gebruik van COBOL snel toe.

In de jaren daarna zijn diverse programmeertalen ontstaan. Toch is momenteel nog een geschatte tachtig procent van de programmeregels in COBOL geschreven.

Het beperkte werkgeheugen leidde ertoe dat zo efficiënt mogelijk moest worden geprogrammeerd. Hierbij werden alle variabelen zo klein mogelijk gehouden, waarbij in datumregistraties de jaren werden beperkt tot de laatste twee posities. Het weglaten van de eerste twee posities leverde een ruimtebesparing op van twee bytes.

Ook in COBOL waren voorzieningen getroffen om het werk met een zo klein mogelijk geheugenbeslag uit te voeren. De datumroutines die destijds standaard werden meegeleverd, gebruikten opnieuw zo min mogelijk geheugen.

Datumvelden komen in circa acht procent van de programmeregels voor en tot zo'n tachtig procent van de programma's bevat datumberekeningen. Gemiddeld wordt geschat dat een bedrijf 20.000 programma's in gebruik heeft, met een gemiddelde omvang van 2000 programmeregels.

Naar schatting is slechts vijftig procent van de datumroutines als zodanig herkenbaar. In de andere helft van de gevallen zijn de datumvelden met cryptische namen weergegeven. Zo kan bijvoorbeeld een datumveld in zes posities in een andere routine worden gedefinieerd met als gevolg dat alleen het jaar wordt verplaatst.

In 1993 hebben zich de eerste datumproblemen voorgedaan bij TANDEM-computers. Evenals in computers van andere leveranciers werd in de computers van de CLX-serie (TANDEM) de systeemdatum intern vastgehouden door een register periodiek te verhogen. In de CLX-computers was echter het register te smal gekozen.

Op 9 november 1993 was het geheel gevuld met waarden 1 en de daarop volgende ophoging resulteerde in een hoop nullen. Het overflow-bit werd genegeerd. Het systeem rekende de registerwaarde trouw om in een kalenderdatum ergens in 1983, zijnde de geprogrammeerde startdatum.

Daarnaast is het ook nog mogelijk dat als de systeemsoftware de volledige datum verstrekt, de toepassingsoftware het jaartal afkapt op twee cijfers.

Indien in mainframes wordt gewerkt met zescijferige data kan dit consequenties hebben voor andere systemen. Als een modern intelligent werkstation wordt aangesloten op een mainframe dan is dit laatste de bepalende factor.

Het kan bovendien gebeuren dat een datumveld een reële waarde moet hebben en dat de waarde 00 in de jaarposities niet wordt getolereerd. Op een mainframe kan dit leiden tot een ABEND (ABnormal END), waarbij het systeem of een deel daarvan 'plat' gaat.

Daarnaast kan het voorkomen dat de programmeur het bij het schrijven van de programma's niet zo nauw heeft genomen met de principes van zuiver programmeren, waardoor men voor onaangename verrassingen komt te staan. Onjuist gebruik van variabelen komt daarbij geregeld voor. Een datumveld kan (even) worden gebruikt voor de opslag van andere gegevens.

Naar schatting is slechts vijftig procent van de datumroutines als zodanig herkenbaar.

Ook het gebruik van het eeuwende (99) en het eeuwbegin (00) voor andere doeleinden dan voor datumaanduidingen kan problemen opleveren. Bijvoorbeeld als het jaartal 00 voor het vastleggen van levenslange abonnementen wordt gebruikt of jaartal 99 met als betekenis 'alle premies geïnd'.

Een ander probleem vlak na de eeuwwisseling is 29 februari 2000. Het jaar 2000 is namelijk een schrikkeljaar. Er zijn applicaties die hiermee geen rekening houden. In de regel is een schrikkeljaar deelbaar door 4, maar niet door 100. In het jaar 2000 is dit echter wel het geval. Het is daarom verstandig hiermee in de analyse van de systemen rekening te houden.

Mogelijke gevolgen

De gevolgen van de eeuwwisseling op de automatisering hebben zowel betrekking op financiële als op operationele systemen. Ter illustratie volgen enige voorbeelden:

Negatieve pensioenjaren

Stel u bent in 1974 begonnen met werken en u wilt in 2002 weten hoeveel pensioenjaren u hebt opgebouwd. Vermindert u 02 met 74 dan is de uitkomst dat u 72 pensioenjaren zult moeten bijbetalen.

Verlopen creditcards

Een creditcardmaatschappij gaf in 1995 een nieuwe serie creditcards uit met een vervaldatum in 2000 (00). Diverse betaalautomaten waren niet bereid geld te geven aan de houders van deze creditcard of betalingen te accepteren op basis van deze creditcards. Deze automaten beschouwden de creditcards namelijk als ouder dan vijf jaar en dus verlopen.

Backup-routines worden niet of onjuist uitgevoerd

Er wordt een backup-tape aangemaakt met een datum in het jaar 2000, waarop de gegevens van de tape kunnen worden gewist. Het tapemanage-

mentsysteem herkent de datum niet en zet de tape vervolgens klaar om opnieuw te gebruiken.

Magazijnsystemen werken onjuist

Een farmaceutisch bedrijf gebruikt een geautomatiseerd magazijnsysteem dat medicijnen op de juiste plaats in het magazijn plaatst, gesorteerd op houdbaarheidsdatum. Door het systeem worden medicijnen met een houdbaarheidsdatum in het jaar 2000 vooraan gezet. De houdbaarheid van deze medicijnen is het kortst, namelijk houdbaar tot het jaar 00.

Lift stopt met werken

Liften zijn vaak met een veiligheidssysteem uitgerust dat ervoor zorgt dat de lift minimaal eenmaal per jaar wordt gecontroleerd door een monteur. Bij de eeuwwisseling geeft het veiligheidssysteem aan dat de laatste controle te lang geleden is (00 – 99 = –99 jaar). De lift weigert verder te functioneren.

Als afsluiter wordt een voorbeeld gegeven van een rampenscenario, waarin meerdere gevolgen zijn verwerkt. De geschetste situatie heeft betrekking op de bedrijfsvoering bij PTT Telecom. Wanneer de klok middernacht slaat in de nacht van 31 december 1999 op 1 januari 2000 kan, als PTT Telecom zich niet heeft voorbereid op de eeuwwisseling, het volgende gebeuren:

Het overbelaste telefoonnet kan de verkeersstroom niet aan. Kleine storingen, veroorzaakt door foute kalender routines, veroorzaken het plat gaan van telefooncentrales. Onmiddellijk wordt overgeschakeld naar uitwijkcentra, maar ook daar slaat het millenniumprobleem toe. Opgeroepen technici arriveren veel te laat op hun plek omdat het beveiligingssysteem hen niet toelaat doordat hun pasjes zijn verlopen. De liften blijven hangen en storingsmonteurs kunnen niet worden opgeroepen omdat ook het semafoonnet plat ligt.

En hoe nu verder?

Het mag duidelijk zijn dat de eeuwwisseling zal leiden tot problemen. In het gunstigste geval ondervindt de onderneming geen schadelijke gevolgen van de eeuwwisseling, maar in het ergste geval kan het voortbestaan van de onderneming ervan afhangen.

Of de onderneming problemen ondervindt en welke gevolgen deze problemen op de onderneming hebben, zal men moeten vaststellen. De onderneming dient hiervoor een risico-analyse uit te voeren.

Er zijn verschillende redenen aan te voeren die het uitvoeren van een risico-analyse rechtvaardigen, zoals:

- het kwantitatief en/of kwalitatief inzicht in de gevolgen en de mogelijke schade;
- het onderkennen van de te treffen maatregelen en, na implementatie, het bepalen van de effectiviteit van de getroffen maatregelen.

Veelal is het inschatten van de kans van het optreden van een bedreiging een problematische zaak. Bij het millenniumprobleem is dit echter niet het geval. Voor de datumroutines die geprogram-

meerd zijn in zes posities staat de zogenaamde 'trigger' vast, namelijk de eeuwwisseling.

Het bepalen van de eventuele schade wordt over het algemeen wat eenvoudiger geacht. Bij het millenniumprobleem gaat dit echter maar ten dele op. De directe schade is eenvoudig vast te stellen, zoals de kosten verbonden aan het eventueel aanpassen van de desbetreffende datumroutines.

Een schade die nagenoeg niet is in te schatten is de indirecte of gevolgschade die kan ontstaan bij het eventueel uitvallen van de geautomatiseerde gegevensverwerking. Er zijn te veel onzekerheden en afhankelijkheden die het schatten onmogelijk maken.

Het object van risico-analyse is duidelijk weer te geven, namelijk alle informatiesystemen, zowel financieel als operationeel. Van deze informatiesystemen dient elke programmaregel te worden beoordeeld.

Problemen waar men, naast het onjuiste gebruik van datumvelden, tegen aan kan lopen bij het uitvoeren van de risico-analyse zijn:

- ontbreken van broncodes (sources/sourcecodes);
- ontbreken van documentatie;
- achterstallig onderhoud;
- slecht programmabeheer.

Begin jaren negentig begon in de Amerikaanse pers het millenniumprobleem de nodige aandacht te krijgen. Veel automatiseringsbedrijven zijn in die tijd begonnen met het ontwikkelen van hulpmiddelen waarmee programma's met datumberekeningen kunnen worden geanalyseerd en – in sommige gevallen – gecorrigeerd. Soms kan het tool ook aangeven hoe datumvelden worden gebruikt, namelijk in berekeningen, in definities of bijvoorbeeld in scherm-layouts.

Het merendeel van de tools is geschreven voor de taal COBOL op het besturingssysteem MVS, wat nu eenmaal bij de banken en verzekeringsbedrijven de meest voorkomende combinatie is. Voor andere hardwareplatformen en talen zijn op de markt minder tools beschikbaar.

Een deel van het millenniumprobleem kan worden opgelost door het gebruik van een aantal standaard conversiehulp programma's. Met behulp van deze conversiesoftware kunnen, afhankelijk van de programmeertaal, de standaard datumroutines worden aangepast. De afwijkende datumroutines en de routines in programmeertalen waarvoor geen conversieprogramma's beschikbaar zijn dienen nog steeds met de hand te worden aangepast. Het aantal aanpassingen dat vervolgens moet worden uitgevoerd, is te veel voor het reguliere personeel, waardoor een extra vraag naar programmeurs op de arbeidsmarkt zal ontstaan. In de jaren dat de programma's moeten worden aangepast (1997 en 1998), wordt verwacht dat de personeelsformaties op de automatiseringsafdelingen tijdelijk met circa dertig procent moeten worden uitgebreid.

Een aantal grote automatiseringsfirma's (CMG, Cap Volmac, Origin en Ordina) is al dan niet geza-

menlijk een actie gestart voor het opleiden van programmeurs.

Daarnaast wordt gekeken naar de mogelijkheden om programma's in het buitenland (met name in India) aan te laten passen. Dit is door taalverschillen vaak alleen mogelijk voor programma's die volledig in het Engels zijn ontwikkeld.

Mondiaal werden de kosten van de aanpak van het millenniumprobleem voor vier (1996 t/m 1999) jaren geschat op circa \$ 600 miljard.

Omgerekend dient per programmaregel rekening te worden gehouden met gemiddeld één dollar, dus ongeacht of de programmaregel dient te worden aangepast.

Voor dit bedrag ontvangt de gebruiker echter geen extra functionaliteiten. Hij investeert alleen in het voortbestaan van zijn onderneming.

RISICO-ANALYSE

Voor het uitvoeren van een risico-analyse is het aan te bevelen vooraf een planning op te stellen voor de fasen voorbereiding, uitvoering en evaluatie.

Vorbereiding

Om de Jaar 2000-problematiek op een succesvolle wijze aan te pakken is het van belang dat het management van de onderneming de noodzaak van het probleem onderkent en middelen beschikbaar stelt voor de probleemaanpak.

In de voorbereidingsfase dient te worden nagedacht over de wijze waarop men het millenniumprobleem denkt aan te pakken. In de op te stellen planning wordt een uitsplitsing gemaakt naar de verschillende fasen van de risico-analyse, waarbij tevens een inschatting wordt gemaakt van de benodigde middelen per fase. Onder deze middelen vallen de kosten die de verschillende werkzaamheden met zich meebrengen en de tijdsinspanning die benodigd is. Het betreft hier niet alleen de benodigde tijd van de (project)organisatie, maar ook de computertijd die beschikbaar moet zijn indien tot aanpassing van de systemen wordt overgegaan. Aangezien de onderneming vaak niet over de benodigde computer-resources beschikt, dient te worden nagedacht over waar de ontbrekende capaciteit vandaan moet komen.

Bovenstaande zaken gelden niet alleen voor informatiesystemen, maar hebben ook betrekking op operationele systemen (waaronder liftcontrolesystemen en klimaatbeheersingssystemen), hardware en systeemsoftware. Zowel bij de systemen als bij de hardware dient aandacht te bestaan voor mainframe-, midrange- en PC-omgevingen.

Uitvoeren impactanalyses

Naast het inventariseren van de systemen en het

bepalen van het bedrijfsrisico dient met de aard van de systemen rekening te worden gehouden.

Inventariseren van de systemen

Om te kunnen bepalen welke applicaties of delen daarvan voor aanpassing in aanmerking komen, dient een inventarisatie te worden uitgevoerd van de binnen de organisatie aanwezige systemen. Er wordt hiervoor een lijst opgesteld met alle in productie zijnde systemen, waarbij wordt aangegeven of het standaardapplicaties of maatwerksoftware betreft. Met behulp van deze lijst wordt aangegeven welke operationele processen worden ondersteund door welke systemen en gegevensbestanden. Bovendien dient per object de verwantschap te worden aangegeven ten aanzien van zaken als: source-codes, JCL's, bestandsbeschrijvingen, SQL-schema's, ontwikkel- en productietools, procedures en documentatie, alsmede het hardwareplatform waarop en de systeemsoftware waaronder de applicatie draait.

Ook van de gegevensbestanden wordt aangegeven door welke applicaties zij, al dan niet gezamenlijk, worden gebruikt. Hetzelfde geldt, per systeem, voor de ingaande en uitgaande gegevensstromen.

Daarnaast kan de organisatie een aantal systemen bezitten die op het moment van inventarisatie niet operationeel zijn. Het is van belang voor elk van deze systemen te bepalen of operationeel gebruik weer aan de orde zal komen. Indien dit het geval is dienen zij meegenomen te worden in de inventarisatie.

De kwaliteit van de medewerkers is een belangrijke factor.

Waarborgen voor de volledigheid kunnen op diverse wijzen worden ingebouwd. Enerzijds dient een inventarisatie van systemen te worden uitgevoerd bij de diverse procesverantwoordelijke managers binnen de organisatie. Deze inventarisatie dient te worden afgezet tegen de geregistreerde systemen bij het rekencentrum. De systemen die centraal worden beheerd, komen vanuit de matching naar voren. Verschillen tussen de registraties kunnen op een eenvoudige wijze worden gecorrigeerd. De systemen die vanuit de inventarisatie en de matching overblijven, dienen betrekking te hebben op end-user computing. Voor met name deze systemen is het van belang dat de inventarisatie juist en volledig wordt uitgevoerd. Dit houdt in dat de kwaliteit van de bij de inventarisatie betrokken medewerkers een belangrijke factor is.

Bepalen van het bedrijfsrisico

Nadat de inventarisatie van de systemen heeft plaatsgevonden, dient een analyse te worden uitgevoerd, waarin het bedrijfsrisico wordt bepaald. De verschillende bedrijfsprocessen dienen te worden geassocieerd naar primaire en secundaire processen.

Aan de hand van het systeemoverzicht kan worden bepaald welke bedrijfsprocessen worden verstoord

als een systeem uitvalt ten gevolge van de eeuwwisseling. Het bestaan van het millenniumprobleem in de systemen kan worden vastgesteld door het uitvoeren van een globale beoordeling op de data. Het is zinvol om hiervoor bijvoorbeeld een simulatietest uit te voeren. Hierbij kan bijvoorbeeld een factuur worden ingevoerd met een factuurdatum in de volgende eeuw. Dan zal blijken of er een verstoring optreedt.

De impact van de (eventuele) verstoringen dient te worden vertaald naar de totale bedrijfsvoering binnen de organisatie.

De uitgevoerde analyse zal informatie opleveren van belang voor het onderkennen van:

- de ernst van het millenniumprobleem;
- het bedrijfsrisico als de problemen niet worden opgelost;
- de prioriteiten welke worden gegeven om bepaalde risico's terug te dringen.

Bepalen van de aard van de systemen

De systemen die problemen ondervinden van de eeuwwisseling zijn geïdentificeerd en de gevolgen van het niet aanpassen van de systemen zijn bepaald.

Er kan nu een keuze worden gemaakt welke systemen dienen te worden aangepast of dienen te worden vervangen en welke systemen Jaar 2000-proof zijn. Hierbij dient onderscheid te worden gemaakt tussen standaardsoftware en zelfbouwsystemen.

In het eerste geval dient contact te worden opgenomen met de leverancier van de software. Er dienen afspraken te worden gemaakt omtrent de eventuele aanpassing van de huidige release. Het kan hierbij voorkomen dat de leverancier van plan is alleen de nieuwe(re) releases aan te passen. Is dit niet de release die binnen de desbetreffende organisatie wordt gebruikt, dan dient te worden nagegaan of de leverancier op basis van het (eventuele) onderhoudscontract kan worden gehouden aan aanpassing van de huidige release.

Is dit niet mogelijk dan dient een keuze te worden gemaakt tussen de huidige release zelf aanpassen of overgaan op een nieuwe wel aangepaste release. Hetzelfde geldt ook ten aanzien van de in gebruik zijnde hardware en systeemsoftware. Aanbevolen wordt de leverancier een verklaring te laten afgeven, dat de aan te schaffen systemen Jaar 2000-proof zijn.

Indien men overgaat tot het zelf aanpassen van systemen dient men wel over de sources te beschikken.

Ook voor de zelf ontwikkelde applicaties dient, indien zij niet op korte termijn moeten worden vervangen, de afweging te worden gemaakt: aanpassen of alsnog (vervroegd afschrijven en) vervangen. In het geval van vervanging geldt evenals bij de aanpassing dat de deadline, namelijk 1 januari 2000 (0:00 uur), vast staat en dat problemen al op een eerder tijdstip kunnen ontstaan. Bij vervanging kan het zowel een nieuw te ontwikkelen applicatie als de aanschaf van standaardsoftware betreffen.

Het aanpassen van de systemen gelijktijdig met het reguliere onderhoudsproces is de meest efficiënte manier om het millenniumprobleem aan te pakken. Door de onderlinge afhankelijkheid van diverse systemen is dit echter niet altijd mogelijk. In dat geval dienen clusters van systemen gelijktijdig te worden aangepast.

Om het aanpassingsproces inzichtelijk te houden is het echter aan te bevelen om in dit proces geen nieuwe functionaliteiten mee te nemen.

Evalueren uitkomsten

De mogelijke oplossingen voor het millenniumprobleem bij de desbetreffende systemen dienen, na het uitvoeren van de impactanalyses, te worden bevestigd.

Om het probleem handmatig te kunnen oplossen, dient de omvang van het aantal aanpassingen voor de organisatie beperkt te zijn. Ondanks het aantal, veelal eenvoudige, aanpassingen en het beperkte aantal personen dat bij de oplossing betrokken is, dreigt een handmatige correctie ook dan nog te duur, te tijdrovend en te foutgevoelig te zijn voor een tijdige, kwalitatieve oplossing.

Een te overwegen alternatief voor de organisatie is het gebruik van een aantal tools om het proces te automatiseren. De meeste bedrijven hebben niet de capaciteit (mensen, tijd en middelen) om deze tools zelf te ontwikkelen. Voor deze organisaties en de organisaties die op basis van een kosten/batenanalyse afzien van de zelfbouw van tools is het te overwegen gebruik te maken van de tools van leveranciers.

Indien er van tools gebruik wordt gemaakt, is het aan te bevelen te kiezen voor de tools die een zo breed mogelijk oplossend vermogen hebben.

Aandachtspunten hierbij zijn:

- de mate waarin het tool verschillende programmeertalen ondersteunt, alsmede de mate waarin het tool het proces ondersteunt (oplossend vermogen);
- de mate waarin het tool het proces automatiseert (het aantal handmatige handelingen dat alsnog moet worden verricht);
- de mate waarin het tool aansluit op de ontwikkelingen binnen de organisatie;
- de mate waarin het tool door de leverancier wordt ondersteund.

Opzetten pilot-projecten

Nadat is bepaald op welke wijze het millenniumprobleem wordt aangepakt, dienen één of meer pilot-projecten te worden opgezet. Door middel van deze pilot-projecten kan worden beoordeeld of de te kiezen oplossingsmethoden en tools de verwachte resultaten opleveren. Daarnaast kan ervaring worden opgedaan met de mogelijke niet ingeschatte problemen waar men tegenaan kan lopen en de inspanning die door de organisatie en eventuele leveranciers van tools dient te worden geleverd.

Het is daarom van belang dat een pilot-project representatief en voldoende ruim wordt gedefinieerd. De pilot dient een aantal voor de organisatie representatieve systemen te bevatten. Dreigt de omvang van een pilot te groot te worden dan is het aan te bevelen deze onder te verdelen naar meerdere pilots.

Aan de hand van de uitkomsten van de pilot-projecten kan een inschatting worden gemaakt van de impact op het totale project.

Aanpassen systemen

Het doel van deze fase is het aanpassen van de systemen zodat deze Jaar 2000-proof zijn. Om dit te kunnen realiseren dient eerst een overall-projectplanning te worden opgezet. Daarna kan de inspanning binnen het project gericht worden op de aan te passen informatiesystemen. Het aanpassingsproces bevat tevens de identificatie van de datumgevoelige velden, aanpassen van de sourcecodes, verifiëren source-code-aanpassing, aanpassingen aan gegevens, testen van gegevens en code en het implementeren van de aangepaste programmatuur en gegevens in de productie-omgeving.

Projectplanning

De planning van het aanpassingsproces dient per project tot in details te worden uitgewerkt. Hierbij dienen de aan te passen systemen te worden onderverdeeld in zodanige clusters dat een verantwoord en beheersbare aanpassing mogelijk is. Deze clusters dienen zaken te bevatten als sourcecodes, copybooks, lists, JCL's en DB-schema's, alsmede gemeenschappelijke (shared) en persoonlijke (uniquely owned) gegevens.

Wanneer het door de omvang van het totale project niet mogelijk is om een beheersbare detailplanning op te zetten, is het raadzaam in ieder geval de planning op te stellen voor de eerste stappen die genomen moeten worden. Gedurende het project dient deze gefaseerde planning aangepast en uitgebreid te worden. Op basis van een netwerkplanning kan de volgorde van de verschillende procesonderdelen worden gerangschikt.

De projectplanning moet zich niet beperken tot het aanpassen van de systemen, maar zal moeten reiken tot het in productie zijn van de aangepaste programmatuur, dus ook inclusief het doorlopen van het testtraject.

Indien systemen moeten worden vervangen, is het nodig in de planning rekening te houden met de afbouw van de oude systemen, het eventueel bouwen, testen en implementeren van de nieuwe systemen, met mogelijke problemen in de aansluiting tussen aangepaste en niet-aangepaste systemen, alsmede de conversie van bestanden van de oude naar de nieuwe programmatuur.

Ook de aanpassing van de te converteren bestanden dient in acht te worden genomen.

Identificeren datumgevoelige velden

Om de aanpassing succesvol uit te kunnen voeren is het van belang de volledigheid van de datumge-

voelige velden vast te stellen. Het betreft hier de velden die aangepast dienen te worden om het systeem Jaar 2000-proof te laten zijn. Dit betekent niet dat alle datumvelden als datumgevoelig worden aangemerkt.

Doordat veel datumvelden zijn opgeslagen met afwijkende namen is het identificeren een moeilijke klus. Niet-geïdentificeerde datumgevoelige velden dienen te worden gezien als tikkende tijdbommen. Het niet juist identificeren kan leiden tot het ten onrechte aanpassen van niet-datumgevoelige systemen en bestanden.

Niet-geïdentificeerde datumgevoelige velden dienen te worden gezien als tikkende tijdbommen.

Het identificeren kan op een aantal manieren geschieden. Doordat de verschillende methoden vaak niet dekkend of haalbaar zijn, zullen combinaties noodzakelijk zijn. Hieronder staat een aantal identificatiemethoden beschreven.

Handmatige identificatie

De verschillende systemen en bestanden worden aan de hand van de code handmatig doorgenomen. De geïdentificeerde datumgevoelige velden worden opgenomen in een tabel. Deze methode is tijdrovend en foutgevoelig. Dit laatste is zeker van toepassing als de handmatige identificatie wordt uitgevoerd door verschillende personen. Met name interpretatieverschillen kunnen leiden tot het onrecht niet identificeren van datumgevoelige velden. Daarnaast kunnen verwijzingen naar andere bestanden en systemen over het hoofd worden gezien.

'Tool'-identificatie

Voordat een tool wordt gebruikt, dient eerst de juiste en volledige werking van het tool zelf te worden vastgesteld. Indien hierover geen zekerheid bestaat kan niet zonder meer op de uitkomsten van het tool worden gesteund. De tools kunnen de identificatie op verschillende manieren uitvoeren, waaronder:

– 'Scanning'. Deze methode zoekt binnen het opgegeven bereik naar datumvelden of daarop lijkende patronen (synoniemen). Deze datumgerelateerde velden hebben over het algemeen een omvang van twee tot tien bytes. Het scanning-programma doorloopt de code en noteert de datumgevoelige velden in een lijst. De aanpassing van de velden dient vervolgens vrijwel altijd handmatig te geschieden. Het identificeren van bepaalde patronen levert over het algemeen niet het gewenste resultaat op. Vaak zijn de synoniemen welke gebruikt worden als zodanig al binnen de organisatie bekend, waardoor hierop al gericht kan worden gezocht. Het detecteren van velden tussen de twee en de tien bytes levert een veel grotere output op dan nodig, waarbij relaties tot datumvelden niet zijn aangegeven. Vervolgens dient deze output handmatig te worden doorlopen op datumgevoelige synoniemen en daarbinnen op relaties met andere da-

tumgevoelige velden. Daarnaast kunnen scanning-programma's geen berekeningen identificeren die op basis van datumvelden worden verricht. Scanning kan over diverse systemen heen worden gedaan. Hierbij worden relaties tussen de systemen niet herkend.

– 'Parsing'. Hierbij worden de relaties tussen de verschillende datumgevoelige velden geïdentificeerd. Het identificatieproces van deze methode maakt eerst gebruik van scanning om de datumgevoelige velden te herkennen, waarna de relaties tussen de verschillende velden kunnen worden herkend. Parsing-programma's herkennen niet de relaties tussen gemeenschappelijke velden en de datumgevoelige velden die worden gebruikt bij I/O-handelingen, zoals communicatiekanalen met randapparatuur.

– 'Propagation'. Deze methode vult die van de parsing aan en werkt met een zogenaamde 'flow-tracing'-techniek, waarbij meerdere programma's gelijktijdig worden doorlopen en de doorwerking van de datumberekeningen door de programma's wordt gevolgd. Om deze methode efficiënt uit te kunnen voeren dienen de verschillende formaten van de datumgevoelige velden bekend te zijn.

De methode van propagation zal het hoogste oplossende vermogen hebben. Doordat deze methode de relatie tussen de datumvelden meeneemt is de kans het grootst dat de synoniemen worden meegenomen. Het is echter wel van belang de juiste werking van de propagation-methode van de gekozen tools vast te stellen.

Aanpassen datumgevoelige velden

Wanneer de datumgevoelige velden inclusief synoniemen en mogelijke relaties en afhankelijkheden zijn geïdentificeerd, kan begonnen worden met de aanpassing van de source-codes. Het aanpassen kan op diverse manieren gebeuren, waaronder een aantal kortetermijnoplossingen. Deze oplossingen zijn vaak sneller te realiseren, maar verschuiven het probleem in plaats van het op te lossen. Hieronder staan enkele oplossingsmethoden beschreven.

Uitsteloperatie

Bij deze methode wordt een nieuwe 'eeuwgrens' gekozen, bijvoorbeeld 2025. Vervolgens worden jaartallen groter dan 25 gezien als een tijdstip in de huidige eeuw. Jaartallen kleiner dan 25 worden gezien als zijnde in de volgende eeuw. Bij het gebruik van deze methode kunnen problemen ontstaan, zoals:

- mensen geboren tussen 1900 en 1925 worden gezien als geboren tussen 2000 en 2025. Deze personen hebben een geboortedatum in de toekomst;
- wanneer een hypotheek wordt afgesloten met een looptijd van dertig jaar zal deze aflopen in het jaar 2026. De computer zal dit echter zien als het jaar 1926, dus als een hypotheek met een negatieve looptijd.

Er kan worden gekozen voor een eeuwgrens van bijvoorbeeld 2050. De situatie als met de hypotheeklooptijd zal voorlopig niet voorkomen, maar

er zullen meer problemen ontstaan met betrekking tot de jaren 1900 tot 1950.

Deze methode brengt op korte termijn relatief niet veel werk met zich mee. In principe elk programma wordt uitgebreid met een module die de gewenste omzetting verzorgt. Doordat in de loop van de jaren de 'nieuwe' eeuwgrens steeds zal moeten opschuiven, is deze methode nogal onderhoudsgevoelig. Het is daarom aan te bevelen deze methode alleen te gebruiken als het desbetreffende systeem binnen afzienbare tijd wordt vervangen, waarbij aandacht dient te bestaan voor mogelijke problemen bij de bepaling van de 'nieuwe' eeuwgrens.

Compressie

Door de meeste applicaties wordt een datum opgeslagen in de vorm van cijfers (in ASCII of EBCDIC). Het is mogelijk de ruimte voor zes cijfers te gebruiken voor het opslaan van acht cijfers. Ieder cijfer krijgt daarbij zes bits tot zijn beschikking, hetgeen voldoende is (totaal 48 bits).

Hexadecimale notatie

Bij deze methode wordt gebruik gemaakt van de hexadecimale notatie. Zo wordt het jaar 2000 weergegeven als A0 in plaats van als 00. Vervolgens wordt het jaar 2010 weergegeven als B0. Het voordeel van deze methode is dat er niet meer ruimte nodig is voor het opslaan van de data en dat ook het sorteren van de data goed blijft verlopen.

Datumveldvergroting

Hierbij moeten alle datumvelden vergroot worden tot acht posities. Deze methode sluit aan op het gebruik van nieuwe datumroutines.

Welke oplossingsmethode ook wordt gekozen, er kunnen communicatieproblemen ontstaan. Zo zal elke applicatie rekening moeten houden met datumformaten en datumtypes, van de interfaces welke worden gebruikt bij de communicatie met andere applicaties, zowel binnen als buiten de organisatie. Hierbij valt bijvoorbeeld te denken aan het elektronisch betalingsverkeer. Indien mogelijk zal dus moeten worden gekozen voor de methode van datumveldvergroting.

Voor het aanpassen van de systemen en bestanden zal de organisatie extra computercapaciteit nodig hebben. De aanpassingen dienen te worden uitgevoerd in een andere computeromgeving dan de productie-omgeving.

Niet alleen de aanpassing van de programma's en bestanden is van belang, maar ook de aanpassing van documentatie waaronder functionele en technische specificaties, alsmede gebruikershandleidingen.

Testen

Nadat de aanpassingen zijn aangebracht, dient te worden vastgesteld of dit juist en volledig heeft plaatsgevonden. Niet alleen de juiste en volledige wijziging van de sourcecodes dient te worden vastgesteld, maar ook de wijziging in bestanden en de

wijze waarop de systemen deze bestanden benaderen (waaronder interfaces).

Daarnaast dient te worden bepaald of er door het aanpassen van de systemen geen negatieve neven-effecten ontstaan.

Er is een aantal tests te onderscheiden, waaronder:

- *systeentest*. Deze test wordt uitgevoerd binnen de afdeling Systeemontwikkeling om vast te stellen dat de aanpassingen juist en volledig zijn uitgevoerd;
- *performancetest*. Door deze test uit te voeren kan worden vastgesteld of de gewijzigde systemen nog draaien binnen de gestelde performance-eisen;
- *integratietest*. De integratietest wordt uitgevoerd om vast te kunnen stellen dat de gewijzigde systemen geen verstoring veroorzaken van de andere systemen;
- *acceptatie-/gebruikerstest*. Door de Gebruikersorganisatie dient te worden vastgesteld dat de gewijzigde programmatuur (nog steeds) functioneert conform de opgestelde functionele eisen. Slechts na deze fase kan op aangeven van de Gebruikersorganisatie sprake zijn van het in productie nemen van de gewijzigde programma's. Voor het uitvoeren van de acceptatietest dient een separate computeromgeving beschikbaar te zijn (acceptatie-omgeving).

Het testen is van toepassing zowel voor de gewijzigde systemen als voor de door leveranciers aangepaste systemen.

Evenals bij het aanpassen van de systemen geldt bij testen dat de organisatie de beschikking dient te hebben over de benodigde extra computercapaciteit.

Indien naar aanleiding van het testen is vastgesteld dat er geen belemmeringen meer bestaan om in productie te gaan, dan kan de Gebruikersorganisatie hiervoor toestemming verlenen.

Evaluatie

Indien bij elke fase de nodige zorgvuldigheid in acht is genomen, kan gerust naar de eeuwwisseling worden toegeleefd. Tenminste voorzover het de organisatie zelf betreft.

Wel dient te worden vastgesteld dat de aan de onderneming gerelateerde bedrijven, waaronder leveranciers en afnemers, het probleem hebben opgelost. Het wegvallen van één van deze schakels heeft wellicht wel schadelijke gevolgen voor de onderneming.

vergelijking tot normale onderhoudsprojecten zijn hier wel bijzonderheden te constateren. Er dient eerst te worden vastgesteld of men daadwerkelijk een probleem heeft. Is dit het geval dan zal het probleem vermoedelijk de gehele organisatie raken. Daarnaast staat de einddatum waarop het probleem moet zijn opgelost vast.

Terugkomend op het probleem 'Is het mogelijk te komen tot een min of meer uniforme aanpak op basis waarvan de millenniumproblematiek op beheersbare wijze kan worden opgelost?' kan worden gesteld dat het antwoord 'ja' is. Ten aanzien van de beheersing dient echter wel te worden opgemerkt dat kwalitatief goed personeel een vereiste is. Dit is met name van belang ten aanzien van de inventarisatie van de systemen, zeker waar het de end-user computing betreft. Voor deze toepassingen in de informatievoorziening kan geen matching plaatsvinden met registraties van systemen waarvan het operationele beheer in handen is van de zogenaamde VO (Verwerkings- en transportorganisatie).

LITERATUUR

[App196] Elaine L. Appleton, *Call in the Cavalry before 2000*, Datamation, January 1996.

[Bark93] P. Barker, *End-of-century problem looms as IS remains idle*, Computing Canada, November 1993.

[Baum96] D. Baum, *Tool up for 2000*, Datamation, January 1996.

[Blan96] H. Blanken, *KPN zet miljoenen opzij voor computerprobleem*, de Volkskrant, mei 1996.

[Buck95] Grant Buckler, *IT begins to face year 2000 PC date problems*, Newsbytes, October 1995.

[Celk91] J. Celko, *Gear up your databases for the 'dreaded' year 2000*, Systems Integration, December 1991.

[Celk96] J. Celko, *Start fixing year 2000 problems now!*, Datamation, January 1996.

[Farb96] A. Farber and R. LaChance, *Time slips away*, Enterprise Systems Journal, February 1996.

[Fine95] D. Fine, *Companies brace for millennium*, InfoWorld, April 1995.

[Grei95] L. Greiner, *Year 2000 a time bomb ready to explode*, Computing Canada, July 1995.

[Hann93] Mary Hanna, *Maintenance burden begging for remedy*, Software Magazine, April 1993.

[Hans96] P.M. Hanselman, *Bedrijfsleven krijgt oog voor eeuwwisseling*, Automatisering Gids, mei 1996.

CONCLUSIE

Bekijken we relativerend de aanpak van het millenniumprobleem, dan kunnen we niet anders dan concluderen dat het hier in beginsel niets anders dan een grootschalig onderhoudsproject betreft. In

O. Venhuis

Is als Consultant/EDP-auditor werkzaam bij Origin/Quality Management Competence Center. Binnen Origin/QMCC is hij verantwoordelijk voor de kwaliteitsbewaking bij millenniumvraagstukken. Daarvoor was hij bij de Concern accountantsdienst van KPN in dienst als EDP-auditor. Hij is afgestudeerd aan de Erasmus Universiteit Rotterdam (postdoctorale opleiding EDP-auditing).

[Heal96] M. Healey, *Het 'millenniumprobleem' speelt al*, Computable, maart 1996.

[Heal96] M. Healey, *Volg de Ansi-regels voor 2000*, Computable, juni 1996.

[Hitc91] Randell L. Hitchens, *Dating problems? Wait 'til the year 2000*, Computerworld, January 1991.

[Houk96] P. Houkes, *De fout van de eeuw*, Haagse Courant, maart 1996.

[Huiz96] J. Huizinga, *De gevolgen van het jaar 2000*, Software Magazine, januari 1996.

[Jage93] P. de Jager, *Doomsday 2000*, Computerworld, September 1993.

[Jage95] P. de Jager, *If you start now ... you just might make it*, Computerworld, November 1995.

[Jage96] P. de Jager, *Eeuwwisseling zet computers op tilt*, Computable, maart 1996.

[Keij95] R. Keijzer, *Ban de bom*, Automatisering Gids, februari 1995.

[Kell89] S. Kelly-Bootle, *The dating game*, Computer Language, July 1989.

[Klim96] T. Klimuc, *Developing a Year 2000 strategy*, American Programmer, February 1996.

[Mart93] Larry W. Martin, *Millennium preparation*, Software Magazine, July 1993.

[McKe95] J. McKendrick, *A once in a century crisis*, Midrange Systems, June 1995.

[McKe95] J. McKendrick, *The year 2000: still way off*, Midrange Systems, October 1995.

[Menk96] B. Menkus, *A strategy for handling the year-2000 problem*, EDPACS, May 1996.

[Praa96] J. van Praat, *Het jaar '2000', je kunt er niet omheen*, De Accountant, juni 1996.

[Rich88] Sheldon L. Richman, *1999 could well be a bonanza year for consultants*, Government Computer News, June 1988.

[Ripp96] R. Rippen, *Wie is bang voor het jaar 00?*, De Financieel-Economische Tijd, juli 1996.

[Scho95] B. Scholtens, *Twee nullen lopen door de Kalverstraat*, de Volkskrant, november 1995.

[Smit95] S. Smith, *The dating game*, Computer Weekly, October 1995.

[Sull95] R. Lee Sullivan, *Ghosts in the machine*, Forbes, June 1995.

[Ulri96] W.M. Ulrich, *Information systems and the Year 2000*, February 1996.

[Varn96] S.E. Varney, *Calm strategy solves year-2000 problem*, Datamation, May 1996.

[Vow195] J. Vowler, *Millennium mayhem*, Computer Weekly, mei 1995.

[Zwar96] C. de Zwart, *Hoogste tijd voor inventarisatie van het probleem*, Informatie Management, maart 1996.

Digital Inform, *Even voorstellen: de 'millennium-bug'*, januari 1996.

The Economist, *Oh what a lovely millennium bug*, August 1996.

Overzicht van overige gebruikte informatiebronnen

Studierapport inzake toepassing van risico-analyse bij geautomatiseerde gegevensverwerking (NGI/mei 1991).

IBM Nederland NV, *The Year 2000 and 2-Digit Dates*, Second Edition, March 1996.

Seminar 'Year 2000 Problems and Solutions' (december 1995) door VAC (Value Added Consultancy) in samenwerking met Micro Focus (leverancier Challenge 2000).

Seminar 'Het jaar 2000' (januari 1996) door NOREA in samenwerking met Coopers & Lybrand en BSO/Origin (leverancier van Template 2000).

Congres 'Het jaar 2000: uw computersystemen van slag' (mei 1996) door Euroforum in samenwerking met Origin, Cap Volmac, CMG, Computer Associates, IP/Software Control en VAC.

HET WERK AFGEMAAKT!

*Prof. mr. dr. J. M. Smits
Hoogleraar Recht en Techniek TU Eindhoven, Adviseur
KPMG*

Wist u dat de gevestigde belangen in de informatie-industrie, denk hierbij aan uitgevers maar ook aan informatie- en telecommunicatiebedrijven, óns er toe zouden kunnen brengen te betalen voor het citeren van cijfers? Namelijk wanneer die cijfers staan voor paginanummers. En de kans bestaat dat cijfers (paginanummers) door een intellectueel eigendomsrecht beschermd zullen gaan worden.

Laat ik dit eens even uitleggen. De bestaande puzzelwoordenboeken voldeden midden jaren tachtig niet goed genoeg meer. Dus besloot de heer Romme De Grote Van Dale over te tikken, daar ontbrekende woorden en meervoudsvormen aan toe te voegen en deze vervolgens met een slim zoekprogrammaatje op de markt te brengen voor de echte puzzelaar. Nu is een woordenboek (ook al heet dat De Grote Van Dale) niets anders dan een verzameling woorden en dus slechts een (grote) hoeveelheid feitelijke gegevens. Toch verloor Romme de rechtszaak omdat, zo zei de hoogste Nederlandse rechter, 'de Van Dale een eigen oorspronkelijk karakter heeft en het persoonlijk stempel van de maker draagt'. Voor Van Dale gold dit, en dus kon de heer Romme de verstokte puzzelaars niet langer van zijn monnikenwerk gebruik laten maken.

Afgelopen zomer bleek dat er nog meer hobbyisten zijn, die proberen te bezien hoever de intellectuele eigendomsrechten van anderen reiken. Men kope of kopiëre de CD-foongids (op CD-ROM) en kraakt de beveiliging, men voegt een veel slimmer zoekprogrammaatje toe en men heeft een prima telefoonnummer-databank voor heel Nederland, en als je die dan alleen voor jezelf gebruikt is er niet(s) (veel) aan de hand. Maar het is veel leuker om meer mensen er gebruik van te laten maken. De ideale plek daarvoor: het Internet. De hobbyisten die ik hier bedoel heten Vuurwerk, een Haarlemse Internet-toegangsverschaffer. Het door hen vanaf 18 april 1996 aangeboden totale Nederlandse telefoonboek heette WEBTEL. Eind mei viel bij Vuurwerk de dagvaarding van PTT Telecom op de mat, op 26 juni was de rechtszitting en op 10 juli 1996 deed de rechter uitspraak. In drie maanden duidelijkheid (hoewel duidelijkheid). De winnaar: PTT Telecom. Natuurlijk!

Nu gaat het in beide hierboven kort aangeduide zaken niet om literatuur of een of andere oorspronkelijke creatieve uiting. Dergelijke uitingen worden in de Nederlandse Auteurswet 'werken' genoemd. 'Werken' hebben recht op bescherming door het auteursrecht, daarover bestaat vooralsnog vrijwel wereldwijd consensus. Het ligt anders in geval van lijsten, feitelijke gegevens, woordenboeken, etc. Derhalve typisch gegevens die uit een databank komen. De Nederlandse wet kent een minder vergaande vorm van bescherming: 'geschriften' die niet oorspronkelijk zijn en openbaar gemaakt of daartoe bestemd zijn (telefoon/woordenboeken),

EDP AUDITORIUM

worden toch nog door de Auteurswet beschermd wanneer er sprake is van directe ontlening.

In de Verenigde Staten speelt op dit moment een zaak die over intellectuele eigendomsrechten op paginanummers gaat. Er is daar eigenlijk maar één uitgever die vrijwel alle rechterlijke uitspraken in de Verenigde Staten publiceert. Dat betekent dat deze uitgever (West Publishing) in vrijwel alle juridische literatuur is terug te vinden. West probeert nu haar monopolie op de (door haar vaak ook nog eens gecorrigeerde) rechterlijke uitspraken c.q. verwijzingen te behouden. (Zie voor veel meer achtergrond <http://essential.org/cpt/legalinfo/legalinfo.html> en <http://www.hyperlaw.com>).

In augustus 1996 is er binnen de WIPO (World Intellectual Property Organization) op mondiaal niveau overeenstemming bereikt over een concepttekst voor een verdrag dat paginanummers, telefoongidsen, eigenlijk alle feiten die op enigerlei wijze verzameld in een (al of niet elektronische) databank zijn opgenomen, beschermt. En dat is natuurlijk vooral in het voordeel van die partijen die het verzamelen van die feiten al heel lang doen, vaak zelfs met publieke middelen en beschermd door wet- en regelgeving. Als u het niet wil geloven, lees: <http://www.loc.gov/copyright/wipo6.html>, en huiver zelf.

De middelen die ons door de Informatie- en CommunicatieTechnologie (ICT)-bedrijven worden aangereikt, maken het ons enerzijds steeds gemakkelijker inbreuk te maken op de rechten van anderen, terwijl tegelijkertijd de vertegenwoordigers van diezelfde bedrijven met argusogen de markt volgen om te bezien of er (eventueel) inbreuk gemaakt wordt op een door hen verworven intellectueel eigendomsrecht.

De ICT degradeert ons vaak tot een nummer, maar het goede nieuws is dat we er nu eindelijk zelf iets over te zeggen krijgen.

CUMULATIEF

1 23e jaargang 96/1

Normbesef
L. Annokée RE en B. Sebregts RE

ISO 9000 en EDP-auditing
Mr. W.R. Nanninga RE en Itkol J.M.W. van de Garde RE

ITIL als inrichtings- en beoordelingsinstrument
Drs. F.J. Hut

De Code voor Informatiebeveiliging
Dr.ir. P.L. Overbeek

De Code voor Informatiebeveiliging als norm voor de EDP-auditor
W.S.C. Krol RE en drs. M.M. Smits

2 23 jaargang 96/2

Besluitvorming over IT-investeringen: gebruik de juiste criteria
Ing. E.M.H. Coorens BE MBA, drs. P.J.C. van Bladel en dr. M. Boogaard

Benchmarking, een hulpmiddel voor de EDP-auditor?
Ir. J.A.M. Donkers RE en mw. ir. E.R. van Sommeren

AS/400-networking
Mw. drs. A.L. Hristova RE

3 23e jaargang 96/3

Systeemsoftware onder controle
Drs. R.H.H.M. Bronzwaer

Fiscale bewaarplicht van gegevens
T.H.C. van de Molengraft RA

System Review Services
Mw. drs. M.J.A. Koedijk RA en mw. W.A. de Munck RA

De EDP-auditor: vertrouwensman, agent of ...?
Drs. H.E. Sijbring RE RA

4 23e jaargang 96/4

Regulering van encryptietechnieken in westerse landen: pro en contra
Mw. A.K.I. Tuinder en mw. mr. A.M.Ch. Kemna MBA

Overheidsaanbestedingen in de IT-branche
Mr. M.J. van Bommel, mw. R. van der Velden en mw. mr. drs. A.W. Duthler

Interconnectie, technisch- en juridisch-relatieve aspecten
Mr. drs. E.F. Clarkson

Bescherming van databases
Mr. A.P. Meijboom

De Wet persoonsregistraties: wet voor buitenstaanders en ingewijden
Mw. prof. mr. J.E.J. Prins

De Europese privacyrichtlijn
Prof. mr. J.M.A. Berkoens

De aansprakelijkheid van de Internet-aanbieder
Mr. S.C. Huisjes

5 23e jaargang 96/5

Corporate governance; de betekenis voor de EDP-auditor
Drs. R.M. Renes RA, prof. dr. P. Wallage RA, J.C. Boer RE RA en ir. J.A.M. Donkers RE

Management control en accountantscontrole – Veranderende inzichten?
Drs. R.M. Renes RA

Het vaststellen van de management-informatie-behoefte; de rol van de tol
Dr. O.C. van Leeuwen RA, drs. R.H.I van Schoubroeck en drs. P.J.A. Kazius

Management control over informatietechnologie
Ir. J.A.M. Donkers RE, drs. R. Oudega RE en ir. J.A. Verstelle RE

6 23e jaargang 96/6

Een beheersbaar verandertraject voor reken-centrummanagers
P. Teeuwen

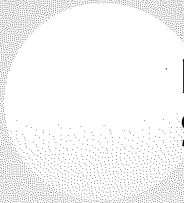
Van rekencentrum naar infocentrum
Dr. ing. H.T.M. van der Zee

Informatiebeveiliging in outsourcing-trajecten
H.R.D. Janus, G. Hulst CISA, ing. A. Shahim en dr. E. Roos Lindgreen

Methoden en technieken van logische toegangs-beveiliging
Drs. ing. E. Beijer

Surfen met de AS/400
Drs. R.Ch.T. Ewals RE

Het einde van Halons
G. Doddrell



KPMG EDP Auditors
Samsom BedrijfsInformatie