

Effectief beheer middels documentatiegeneratie

dr. Arie van Deursen (CWI)
dr. Tobias Kuipers (Software Improvement Group)



©2002 Software Improvement Group B.V.

1 IT Beheer

IT Beheer houdt zich bezig met de instandhouding van software. Essentieel aan deze instandhouding is dat het systeem ten alle tijde aangepast kan worden aan nieuwe requirements of een veranderende omgeving, en dat gevonden fouten snel gerepareerd kunnen worden. Deze eis van evolutie enerzijds en instandhouding anderzijds is een van de grootste uitdagingen van IT Beheer. In dit artikel gaan we in op de rol die documentatie, en in het bijzonder gegenereerde documentatie, kan spelen in IT Beheer ter ondersteuning van de noodzakelijke evolutie van het beheerde systeem.

De moderne visie op IT Beheer benadrukt het adequaat inspelen op (1) de eisen die door het gebruik van informatiesystemen worden gesteld, (2) de randvoorwaarden die door het management worden gedictieerd, en (3) de externe invloeden van bijvoorbeeld bedrijfseconomische of bestuurlijke aard (Looijen, 1999). Het beheer zelf omvat de instandhouding van de apparatuur, programmatuur, gegevensverzamelingen en procedures, benodigd om de door een informatiesysteem gerealiseerde gegevensverwerkings- en informatievoorzieningsprocessen te waarborgen.

Eisen die aan operationele informatiesystemen of afzonderlijke componenten gesteld kunnen worden richten zich op een scala aan aspecten (Looijen, 1999) waarvan in dit artikel de volgende naar voren zullen komen:

- Betrouwbaarheid van de componenten waaruit een informatiesysteem is opgebouwd: Mate waarin de gebruiker van een informatiesysteem kan vertrouwen op de juistheid, volledigheid, tijdigheid, en geoorloofdheid van de gegevensverwerking en informatievoorziening;
- Controleerbaarheid van verwerking en wijziging: Mate waarin de gegevensverwerkings- en informatievoorzieningsprocessen en de wijzigingen binnen informatiesystemen beheerd verlopen;
- Flexibiliteit van informatiesystemen met betrekking tot aanpassingen: Mate waarin uitbreidingen of variaties op informatiesystemen zijn aan te brengen;
- Onderhoudbaarheid van informatiesystemen: Mate waarin de informatiesysteemcomponenten zijn aan te passen als gevolg van te plegen onderhoud.

IT Beheer vindt plaats onder door het management gedefinieerde randvoorwaarden. Hiertoe behoren onder meer de totale beschikbare financiële middelen en standaardisatierichtlijnen om zodoende het IT Beheer in de gehele organisatie effectiever uit te kunnen voeren.

2 Software-evolutie

IT Beheer vindt plaats op systemen die waardevol zijn voor de gebruikers van deze systemen. Wil een software systeem zijn waarde behouden, dan dient het continu aangepast te worden, een wetmatigheid die reeds is geformuleerd door Belady en Lehman (1976). De aanpassingen kunnen variëren van kleine correcties, eenvoudige functionele uitbreidingen, wijzigingen om in te spelen op veranderingen in de technische, zakelijke, of juridische omgeving, tot grootschalige uitbreidingen van een bestaand systeem. In dit artikel gebruiken we de term “software-evolutie” voor de specifieke activiteiten waarin een software-systeem daadwerkelijk wordt aangepast (Bennett & Rajlich, 2000).

Het is duidelijk dat evolutie een essentieel onderdeel van beheer vormt: er is immers een sterke wisselwerking tussen IT Beheer en software-evolutie. Enerzijds is de continue evolutie een belangrijke complicerende, en dus kostenbepalende, factor in het beheren van een informatiesysteem. Anderzijds dient IT Beheer de randvoorwaarden voor succesvolle evolutie te scheppen. Hiertoe behoren:

- Adequate afhandeling van change requests.
- Faciliteren van de planning en impact-analyse voor wijzigingen.
- Aanlevering van de juiste versies van de systeemcomponenten, functionele documentatie, ontwerpdocumentatie enz., benodigd om de wijzigingen door te voeren.
- Aanbieding van de juiste gereedschappen om wijzigingen betrouwbaar en effectief door te voeren.
- Bewaking van de kwaliteit van ten gevolge van doorgevoerde wijzigingen aangepaste systeem-onderdelen.

Voor al deze activiteiten is een goed begrip van de structuur van het onderliggende software-systeem noodzakelijk. De meest gangbare schattingen omtrent de omvang van de inspanning benodigd voor het verkrijgen van dit begrip is dat deze op kan lopen tot 50

Wil het IT Beheer software-evolutie op adequate wijze ondersteunen, dan dient het faciliteren van program understanding een belangrijke plaats in de gekozen aanpak in te nemen. Middels goed beheer van onderliggende systeemelementen en bijbehorende documentatie zou het immers mogelijk moeten zijn program understanding effectiever te laten verlopen, en zodoende aanzienlijke besparingen op het gebied van software-evolutie te realiseren. In dit artikel gaan we nader in op deze relatie tussen software evolutie, program understanding en IT Beheer.

3 Documentatiegeneratie

De basis voor betrouwbare evolutie wordt gevormd door gedegen technische documentatie van de software. Meer dan de helft van de tijd besteed aan onderhoud wordt in beslag genomen door pogingen de software te begrijpen. Goede documentatie kan helpen dit begrijpen sneller en beter te laten verlopen.

In de praktijk heeft documentatie echter vaak te lijden onder diverse ernstige problemen:

- Het schrijven van documentatie wordt als vervelend ervaren.
- Het schrijven van documentatie is kostbaar.
- De beschikbare documentatie is niet van het gewenste abstractienivo.
- Documentatie verandert vaak niet mee met het onderliggende systeem, en is daardoor inconsistent en een potentiële bron van fouten.
- De beschikbare documentatie is niet van het gewenste abstractienivo.

- De documentatie van systemen in een organisatie is niet uniform: verschillende systemen hanteren hun eigen documentatie-standaards.
- Documentatie is vaak moeilijk toegankelijk, en niet eenvoudig beschikbaar vanaf een standaardlocatie.
- In de klassieke visie op systeemontwikkeling wordt documentatie opgeleverd bij oplevering van een softwaresysteem. Bij systeemwijzigingen worden alleen de wijzigingen gedocumenteerd. Dit heeft tot gevolg dat er al vrij snel geen integrale, actuele, documentatie van het systeem meer is (Boef e.a., 2001).

De IT Beheerder zal een documentatie-aanpak moeten vinden die deze problemen adresseert.

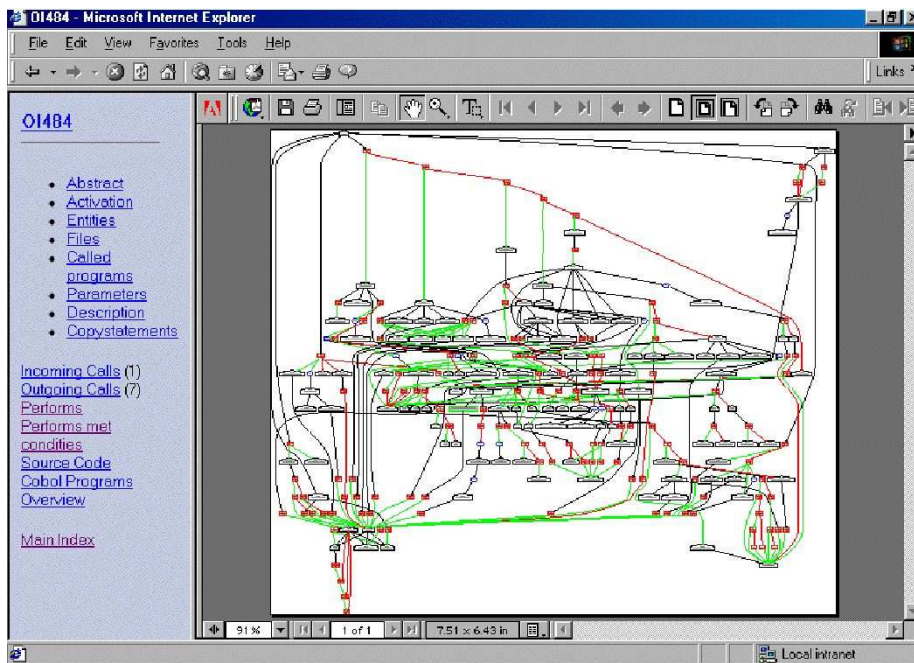
Een nog weinig gebruikte werkwijze is het automatisch genereren van systeemdocumentatie. Hierbij worden zowel individuele programma's als complete programma-portfolios geanalyseerd. Op basis van deze analyse wordt zowel tekstuele als grafische documentatie gegenereerd die via het Intranet toegankelijk gesteld kunnen worden. (Boef e.a., 2001; van Deursen & Kuipers, 1999).

Een documentatiegenerator analyseert programma's, en probeert daar zoveel mogelijk informatie uit te destilleren. Voorbeelden hiervan zijn:

- Door welke programma's een bepaald programma wordt aangeroepen.
- Tekstuele beschrijvingen over het doel van een programma.
- Het gebruik van databases, copybooks, en bestanden.
- Indexering van kernwoorden die in commentaar voorkomen.
- Extractie van business-regels

De afgeleide informatie kan op diverse manieren getoond worden. Het gebruik van hypertext maakt het mogelijk om via links door de informatie te browsen, en tekst en plaatjes te combineren. Van belang hierbij is dat:

- Er zowel hoog-nivo views als gedetailleerde views getoond worden
- Er eenvoudig ingezoomd kan worden van hoog naar gedetailleerd nivo en omgekeerd uitgezoomd van laag naar hoog abstractienivo.
- Er meerdere structuren en afhankelijkheden getoond worden, zoals database en call relaties, copy relaties, enz.
- De relaties tussen de verschillende structuren inzichtelijk gemaakt worden, waarbij bijvoorbeeld van de ene naar de andere "gezapt" kan worden.
- Informatie zowel tekstueel als visueel gepresenteerd wordt. Visuele informatie omvat grafen (voor bijvoorbeeld call afhankelijkheden) en grafieken (voor bijvoorbeeld metriecken). Tekstuele informatie omvat beschrijvingen van functionaliteit, allerlei lijsten van gebruikte structuren, enz. (Zie figuur 1 voor een voorbeeld).
- Zowel browsing als searching adequaat wordt ondersteund.



Figuur 1: Documentatie is zowel grafisch als tekstueel beschikbaar

- Medewerkers de gegenereerde documentatie kunnen verrijken met notities over bepaalde componenten of modules, of met handgeschreven, bijvoorbeeld functionele documentatie.

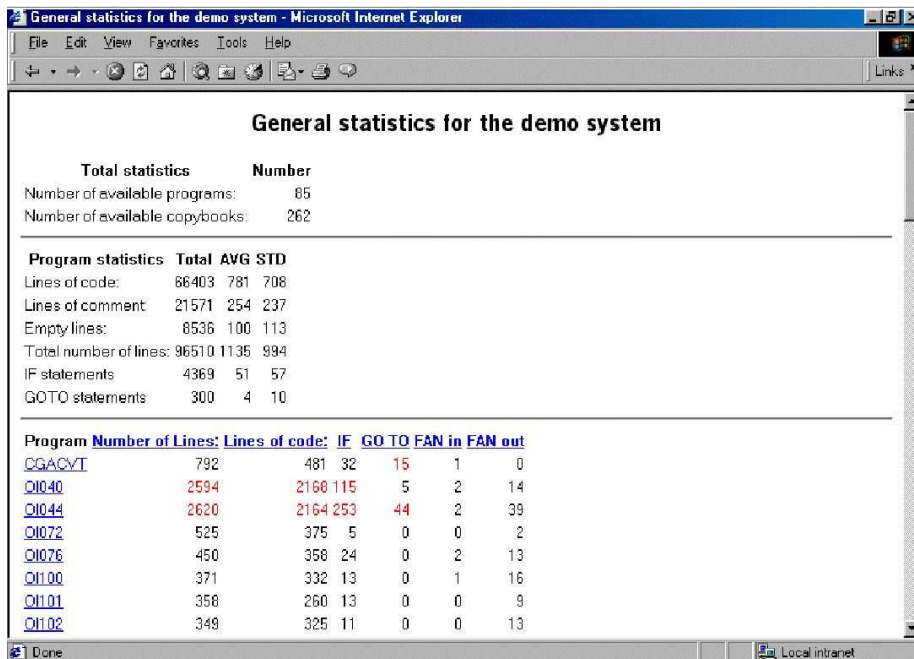
De gevonden informatie kan, afhankelijk van het produkt, op diverse manieren bekeken worden variërend van proprietary browsers tot standaard web-browser zoals Netscape of Internet Explorer.

De getoonde informatie maakt de technische structuur van het software-systeem zichtbaar, om zo de daarin bevatte systeemkennis beter toegankelijk te maken. Deze kennis kan voor diverse doeleinden worden gebruikt, variërend van eerste kennismaking met een systeem (om de inwerktijd van nieuwe beheer-medewerkers te verkleinen) tot meer diepgaande pogingen een specifiek programma of sectie te begrijpen (geschikt voor impact-analyses bij software-evolutie).

De getoonde informatie (zoals bijvoorbeeld metriecken (zie Figuur 2)) kan ook gebruikt worden voor het vormen van een kwaliteitsoordeel over de gedocumenteerde systemen. Men kan hierbij denken aan de intake van nieuwe systemen, bijvoorbeeld bij het aangaan van een outsourcingsovereenkomst. Verder is het mogelijk kwaliteitscontroles geautomatiseerd door te voeren op het moment dat een systeem na wijziging opnieuw wordt aangeboden aan het configuratie-management systeem. De resulterende rapportages kunnen als onderdeel van de gegenereerde documentatie opgeleverd worden.

Bij de keuze en implementatie van documentatiegeneratie kan men zich laten leiden door drie benaderingen die elk op basis van bronprogramma's tot daaruit afgeleide documentatie leiden:

- Standaard commerciele tools met vaste functionaliteit: hierbij zijn geen aanpas-



Figuur 2: Een overzicht van metriekeken

singen mogelijk aan klantspecifieke wensen.

- Dienstverlening waarbij de documentatiegeneratie uitbesteed wordt: hierbij kan de documentatie ook voor een deel handmatig gemaakt worden.
- Aanpasbare documentatiegeneratoren: hierbij is de mate van automatisering het hoogst en kunnen klantspecifieke wensen het eenvoudigst gehonoreerd worden.

Bij de keuze voor één van deze benaderingen moet men vooral letten op de ondersteuning van bedrijfsspecifieke idiomen en de aansluiting op de interne beheerprocessen.

De ideale situatie bij het beschikbaar stellen van documentatie is dat zodra een bronprogramma naar de productie-omgeving overgedragen wordt deze automatisch ook naar een documentatie-server gestuurd wordt. Deze verwerkt het bronprogramma en meegestuurde extra informatie, en stelt de afgeleide documentatie direct beschikbaar via het Intranet. Na de selectie van de documentatiegenerator en eventuele aanpassing daarvan kan een begin gemaakt worden met de feitelijke invoering. Afhankelijk van de omvang (vaak tientallen miljoenen regels) kan in een periode van enkele dagen tot enkele weken de hele software-portfolio automatisch gedocumenteerd worden. Vanaf dat moment is de technische programmadocumentatie actueel. Door de koppeling aan het IT Beheer blijft deze ook actueel. Door de beschikbaarstelling via het Intranet is de toegankelijkheid van de documentatie ten slotte gegarandeerd. Het is duidelijk dat documentatiegeneratie zich met name richt op de technische aspecten van systeemdocumentatie. Wel is het mogelijk om de zo gegenereerde technische documentatie te integreren met handgeschreven functionele documentatie. Op deze manier wordt ook de functionele documentatie via het intranet centraal beschikbaar gesteld. Bovendien

is het mogelijk om geautomatiseerd dwarsverbanden aan te leggen tussen gegenereerde documentatie enerzijds en handgeschreven documentatie anderzijds.

4 Documentatiegeneratie in de praktijk

De voorgestelde werkwijze is reeds op diverse plekken in de praktijk gebracht. In deze sectie kijken we in meer detail naar ABN AMRO, waar geautomatiseerde documentatie sinds 2000 systematisch wordt toegepast. Om de afdeling support te ondersteunen bij het beheer van de Cobol mainframe systemen worden deze automatisch geanalyseerd en gedocumenteerd. Het betreft alle 30.000 Cobol sources van ABN AMRO, met een totale omvang van 50.000.000 regels code.

ABN AMRO heeft hiervoor gebruik gemaakt van de benadering waarbij een standaard documentatiegeneratie tool geconfigureerd wordt ten behoeve van bedrijfsspecifieke eisen en wensen. Zo wordt de call handling faciliteit van ABN AMRO ondersteund, en worden Cobol listings (inclusief copybooks) geanalyseerd in plaats van losse programma-sources. De documentatiegenerator draait bij ABN AMRO op een IBM RS/6000 computer. Via een overdrachtssluis kunnen sources automatisch van het mainframe naar de documentatie server gestuurd worden. Deze sluis is ingebed in de logistieke en beheerprocessen. De overdracht wordt automatisch in werking gezet door het configuratie management systeem van ABN AMRO, elke keer dat een source van de testomgeving naar de productie-omgeving wordt gepromoveerd.

De eerste stap na installatie van de documentatie-server is het uitvoeren van een inhaalslag, om de bestaande sources een eerste keer te documenteren. Deze initiele run nam voor alle sources ongeveer vier dagen in beslag, en heeft de technische documentatie voor alle Cobol mainframe programma's in een klap up-to-date gemaakt. Om deze documentatie actueel te houden is vervolgens de koppeling met het logistieke proces geactiveerd. Op deze manier blijft de documentatie actueel, wat voor ABN AMRO een van de belangrijkste redenen was om geautomatiseerde documentatiegeneratie in te stellen.

De afgeleide documentatie wordt beschikbaar gesteld via het ABN AMRO intranet, en kan vanaf elke werkplek bij ABN AMRO middels een standaard web browser bekeken worden. Op deze manier wordt kennis centraal via het intranet ontsloten, in plaats van in verschillende versies verspreid over diverse mappen in verschillende kasten. Door het gebruik van documentatiegeneratie heeft ABN AMRO een aanzienlijke besparing in de kosten van IT Beheer weten te realiseren. Allereerst is een inschatting gemaakt volgens welke de inhaalslag zelf 15 mensjaar in beslag zou nemen indien deze via de reguliere procedures uitgevoerd zou worden. Het actueel houden van de documentatie zou vervolgens nog een 3 tot 5 mensjaar vergen, kosten die elk jaar terug zouden komen. Deze besparing betekent tevens dat de zo vrijgekomen menskracht ingezet kan worden voor andere beheertaken. Een verdere kostenbesparing wordt gerealiseerd doordat ABN AMRO systemen nu sneller in productie kan nemen, daar immers minder inspanning vereist is voor het documenteren.

Naast kostenbesparingen heeft het gebruik van documentatiegeneratie ook geleid tot een kwaliteitsverbetering. De beheer- en onderhoudsmedewerkers worden beter ondersteund bij het begrijpen van de structuur van software systemen, en kunnen zo hun taken sneller en beter uitvoeren. Nieuwe medewerkers kunnen sneller aan de slag dankzij de beschikbare documentatie. De medewerkers zelf hoeven verder minder tijd te besteden aan het documenteren; als ze dit doen dan kunnen ze zich richten op meer hoog nivo informatie in plaats van op het documenteren van technische details. Door

de uniformiteit van de afgeleide documentatie, ten slotte, is het tevens eenvoudiger geworden mensen te laten werken aan het beheer van meerdere systemen.

De verdere uitbouw van het gebruik van de mogelijkheden van documentatiegeneratie is bij ABN AMRO ingericht als een iteratief proces. In eerste instantie is een “minimale” documentatiegenerator opgezet, om zo ervaring te krijgen met het gebruik van generatie technieken op de schaal die nodig is voor ABN AMRO. Vervolgens worden regelmatig nieuwe releases van de generator ingevoerd. Hierdoor is mogelijk om te leren van gebruikers, en om eventuele knelpunten snel op te lossen. Verder kan op deze manier geprofiteerd worden van het voortschrijdend inzicht op het gebied van reverse engineering en redocumentatie technologie.

5 Samenvatting en Conclusies

Er is een sterke wisselwerking tussen IT Beheer enerzijds en software-evolutie anderzijds. Een beheerd systeem dient te evolueren, wil het zijn waarde voor de organisatie behouden. Om effectieve evolutie mogelijk te maken, dient IT Beheer de juiste randvoorwaarden te scheppen.

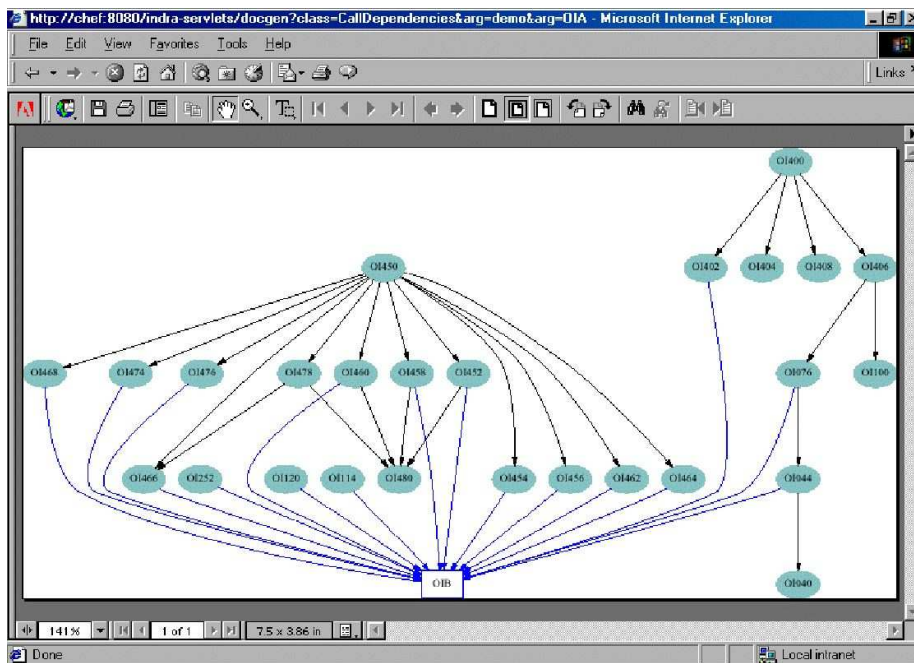
De basis voor betrouwbare evolutie wordt gevormd door gedegen technische documentatie van de software. Meer dan de helft van de tijd besteed aan onderhoud wordt in beslag genomen door pogingen de software te begrijpen. Goede documentatie kan helpen dit begrijpen sneller en beter te laten verlopen.

In de praktijk heeft documentatie echter vaak te lijden onder de diverse ernstige problemen. Zo wordt documentatie vaak slecht bijgehouden, wordt het schrijven ervan als vervelend ervaren, is de documentatie moeilijk te vinden, en niet van het gewenste abstractienivo. Een gevolg is dat de documentatie van onvoldoende kwaliteit is.

Om deze problemen te adresseren stellen we een werkwijze voor waarin documentatie zo veel mogelijk automatisch gegenereerd wordt. Daarbij is het van belang dat het IT Beheer proces zodanig wordt ingericht dat de documentatiegenerator automatisch wordt aangeroepen op het moment dat de programmatuur wijzigt de bijbehorende documentatie opnieuw wordt gegenereerd. Deze werkwijze wordt reeds op diverse plekken succesvol toegepast, onder meer bij ABN AMRO.

Middels de voorgestelde werkwijze kunnen de meeste van de beschreven documentatieproblemen opgelost worden:

- Er hoeft minder documentatie geschreven te worden, wat leidt tot kostenbesparingen.
- De “saaie” documentatie wordt gegenereerd, waardoor de onderhoudsmedewerkers zich kunnen richten op het uitleggen van de onderliggende redenen van beslissingen, in plaats van op de details van de feitelijke beslissingen.
- Handgeschreven documentatie kan meegelinkt worden in gegenereerde documentatie.
- De documentatie kan diverse views op het software-systeem geven, variërend van een hoog-nivo helikopter view (zie bijvoorbeeld figuur 3) voor planning en impact analyse, tot een gedetailleerd code nivo nodig voor het aanbrengen van hot fixes op het moment dat zich acute problemen voordoen.
- Dwarsverbanden tussen verschillende vormen van documentatie kunnen automatisch aangelegd worden.



Figuur 3: Een overzicht van een subsysteem

- Zodra een nieuwe versie van de programmatuur onder beheer wordt geplaatst, wordt automatisch up-to-date documentatie gegenereerd, waardoor de betrouwbaarheid gegarandeerd is.
- Het gebruik van een en dezelfde documentatiegenerator voor verschillende systemen garandeert uniform gepresenteerde documentatie, zelfs als deze in verschillende talen zijn geschreven. Dit maakt het uitwisselen van medewerkers over verschillende systemen mogelijk.
- Een inhaalslag om voor systemen zonder documentatie toch documentatie te verkrijgen kan geautomatiseerd, tegen lage kosten gerealiseerd worden.
- De documentatiegenerator kan de afgeleide documentatie centraal neerzetten, en zo uniform beschikbaar stellen voor alle medewerkers. Op deze manier kan een bijdrage geleverd worden aan het voldoen aan de eisen die gesteld worden aan IT Beheer: gegarandeerde onderhoudbaarheid, flexibiliteit, controleerbaarheid en betrouwbaarheid tegen beheersbare kosten.

Acknowledgements. Dank aan Jan van Bon voor overleg over dit artikel.

Dit artikel is gepubliceerd in:

IT Beheer Jaarboek 2002, het standaardwerk voor IT Service Management. Onder redactie van Drs. Jan van Bon

Uitgegeven door Ten Hagen & Stam Uitgevers, Den Haag, 2002.

Literatuur

(Belady & Lehman, 1976). L. A. Belady and M. M. Lehman. A model of large program development. IBM Systems Journal 15(3):225-252, 1976

(Bennett & Rajlich, 2000). K. Bennett and V. Rajlich. Software Maintenance And Evolution: A Roadmap. In A Finkelstein (ed.) The Future of Software Engineering. ACM Press, 2000.

(Boef e.a., 2001). J. Boef, A. van Deursen en P. Klint. Goede softwarelogistiek basis voor snelle aanpassingen. Automatisering Gids 35(36), 7 september 2001.

(Corbi, 1989). J. Corbi. Program Understanding: Challenge for the 1990s, IBM Systems Journal, 28(2):294-306, 1989.

(van Deursen & Kuipers, 1999). A. van Deursen and T. Kuipers. Building Documentation Generators. In Proceedings of the International Conference on Software Maintenance (ICSM'99), pages 40-49. IEEE Computer Society, 1999.

(Looijen, 1999). M. Looijen. Inrichten en beheren; Totaal inrichtings- en beheer-concept. Hoofdstuk VI.1, ICT-zakboekje. pp. 1143-1174. PBNA, 1999.

Over de auteurs

dr. Arie van Deursen is werkzaam als onderzoeker en projectleider bij het Centrum voor Wiskunde en Informatica (CWI) te Amsterdam. Zijn onderzoeksinteresses liggen op het terrein van program understanding, reverse engineering, en software testing. URL: <http://www.cwi.nl/~arie/>

dr. Tobias Kuipers is hoofd research & development bij de Software Improvement Group, een CWI spin-off bedrijf dat zich richt op het verhelpen van problemen op het gebied van legacy systemen. Hij kan bereikt worden via tk@software-improvers.com. URL <http://www.software-improvers.com>

Reactie redactie IT-Beheer Jaarboek

Uit: IT Beheer Jaarboek 2002, p. 151

Het IT Beheer Jaarboek is een publicatie van de vereniging ITSMF Nederland en wordt uitgegeven door Ten Hagen & Stam Uitgevers.

Reactie van de redactie op artikel Kuipers & Van Deursen:

Kuipers & Van Deursen brengen een zeer interessante technische mogelijkheid rond het documenteren van zogenaamde legacy-systemen onder de aandacht. Documentatie is essentieel voor het beheer van applicaties, daarover zullen de meeste beheerders het wel eens zijn. Zeker als het gaat om modificaties op bestaande applicaties is het van groot belang de toegepaste mechanismen in de software goed te doorgronden en te overzien, alvorens tot het aanbrengen van wijzigingen kan worden overgegaan. Als er dan geen documentatie aanwezig is, leidt dit voornemen tot modificatie (change) maar al te vaak tot tijdrovend en moeilijk speurwerk naar de werking van de applicatie. Een tool waarmee documentatie (automatisch) gegeneerd kan worden brengt in dit soort gevallen vanzelfsprekend op een bijzondere manier uitkomst en kan leiden tot gewel-

dige besparingen en kwaliteitsverbeteringen in de omgeving van de IT-beheerder.

De ontwikkeling van tools zoals hier beschreven brengt potentieel wel enig risico met zich mee. Deze manier van documenteren kan namelijk ook worden ingezet bij de ontwikkeling van nieuwe applicatiesoftware. Het is dan de vraag of de resultaten die met deze tool eenvoudig en efficiënt kunnen worden gegeneerd van hetzelfde kwaliteitsniveau zijn als de resultaten van een goed uitgevoerd traditioneel documentatietraject. Als dat wel het geval blijkt te zijn is het belang van dit mechanisme alleen maar des te groter: het kan leiden tot een substantiele kwaliteitsverbetering van het ontwikkelproces terwijl tegelijkertijd een kostenreductie plaats vindt: een situatie die voor de ontvangende beheerder bijna te mooi is om waar te kunnen zijn.