Profiling, het bepalen en oplossen van performance-flessenhalzen, raakt aan veel terreinen. Het is dan ook geen wonder dat testspecialist Mercury op dit terrein een woordje meespreekt, en de toolset zelfs heeft uitgebreid in de richting van totaal systems- en changemagement. Essentieel daarbij is of de gekozen agent-loze technologische aanpak voldoende profiling-kwaliteiten brengt, en of we hiermee écht alle soorten J2EE en .NET-applicaties aankunnen.

thema

Mercury Performance Center: productieve profiling

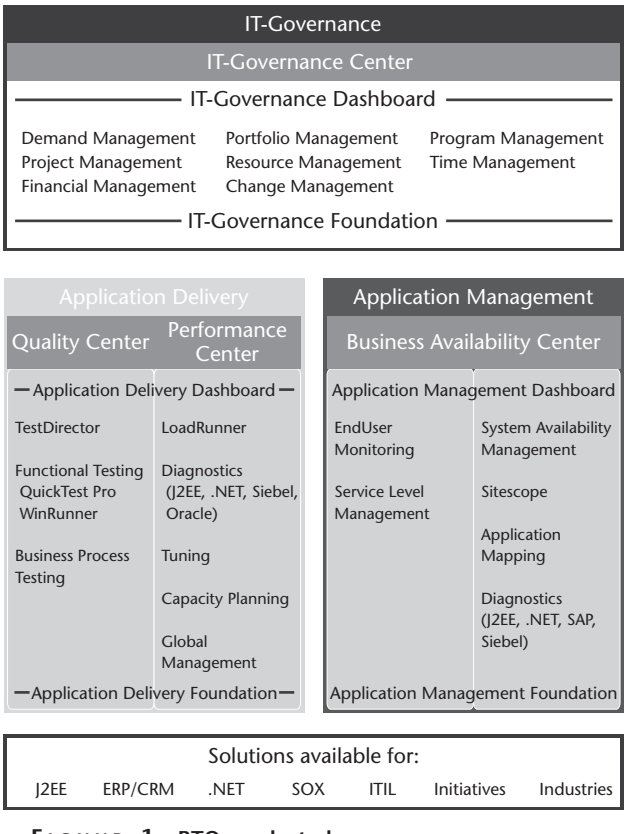
Multi-platform profiling (2)

De tools die we bekijken moeten niet alleen multi-platform zijn, en geschikt voor meerdere programmeertalen, maar ook compleet. Die brede dekking komt neer op drie basiswensen, die in het vorige artikel uit deze serie uitvoerig zijn besproken (zie Software Release Magazine 4/2005):

1. *Breakdown-analyse* van de totale responsetijd binnen de test-omgeving, zo 'diep' binnen de applicatie als maar haalbaar is. Hierbij moet het tool helpen om variërende testomstandigheden te simuleren: bijvoorbeeld minder RAM, of verdubbelde load, of een modemverbinding.
2. *Diepgaande element-measurements* in productie, gerelateerd aan beperkte breakdown-analyses aldaar. Element-measurements zijn de klassieke performancemetingen per machine of zelfs server-taak, en tonen mede of een machine gezond of ongezond belast wordt door onze applicatie. Verder hebben we voor de productie beperkte breakdowns nodig. Het is voldoende om te zien in welke fysieke component (on)gezond veel doorlooptijd verbruikt wordt. Vervolgens moet het tool helpen de relatie te leggen met de elementen; want het ongezonde gedrag kan liggen aan slechte programmering of aan onjuiste machine-sizing.
3. *Trendrapportages* over de productiemetingen, uiteraard met startpunten (baselines).

We kunnen gewapend met deze lijst nu de Mercury-tools de maat gaan nemen.

MERCURY OVERALL-PORTFOLIO Mercury heeft zijn wortels in de testwereld, middels de al generaties aan de kop staande WinRunner- en LoadRunner-tooling. Hoewel LoadRunner altijd al productiemetingen aankon, is de inzet van deze techniek nu behoorlijk uitgebreid tot een totale diagnose-omgeving. Ook monitoring op afstand (de Topaz-lijn), al een tijd aanwezig, is uitgebreid. Een belangrijke verbreding vond in 2003 plaats toen het, eveneens in Silicon Valley gevestigde,



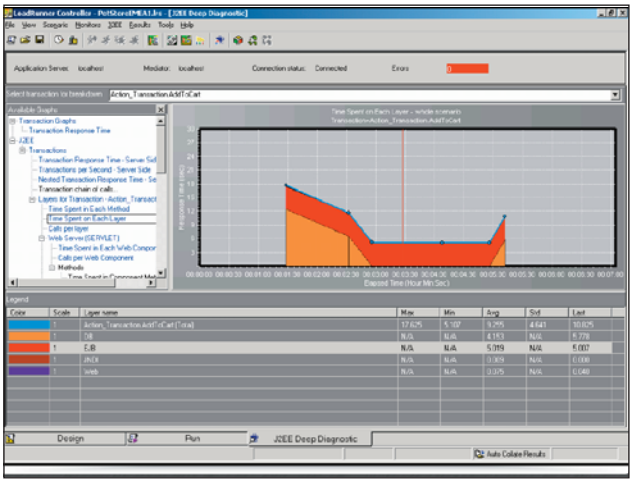
FIGUUR 1. BTO-productschema.

Kintana werd overgenomen. Vanuit deze bloedgroep werd project- en changemanagement binnengebracht. Veel van de producten worden ook als MSP-service (Management Service Provider) aangeboden. Men gebruikt als overkoepelende aanduiding BTO (Business Technology Optimization). In deze categorie vinden we ook tools van onder meer Compuware, CA, Borland en IBM. Het portfolio wordt samengevat in Figuur 1.

We zien hier drie categorieën en vier 'centers'; de Application Delivery-categorie kent als enige twee centers. Een categorie wordt door Mercury gedefinieerd als een fase binnen de applicatie-levenscyclus, ook terug te vinden in indelingen als ITIL. En zo'n fase krijgt een eigen 'dashboard'-console, zoals bijvoorbeeld het Delivery-dashboard; en een gemeenschappelijke 'betonlaag' die praat met de configuratie-database (CMDB).

Een center is veel meer een groepering van functies voor een bepaalde doelgroep. Vandaar dat binnen Delivery twee centers bestaan: die voor de functionele applicatiekwaliteit en die voor de performance. En sommige functionaliteit kan binnen meerdere fasen nodig zijn. Het duidelijkste voorbeeld is het tool Mercury Diagnostics, dat zowel in testomgevingen nodig is als voor 'Application Management' in productie.

Vanuit onze profiling-wensen is nu ook direct duidelijk welke Mercury-centers onze aandacht hebben. Dat betreft Performance maar ook BAC (Business Availability Center). Onder de laatste groep vallen namelijk alle productiemetingen, dus ook responstijd en element measurements. Over de andere twee centers geven we dus slechts kort informatie. Het Quality Center betreft zoals we zien de functionele kwaliteit. WinRunner en QuickTest zorgen voor automatische tests met screenscraping-achtige technieken, TestDirector is een testmanagement-oplossing (inclusief workflow) voor het hele proces en Business Process Testing maakt de testfuncties toegankelijk voor business-mensen met scripting-aversie.



FIGUUR 2. Voorbeeld van Diagnostics.

Het IT Governance-gebeuren is echt gericht op projectmanagers en IT-bedrijfsbureaus. De hele bouw-keten (inclusief tests) kan vanaf topniveau tot in applicatiedetails worden bekeken en aangestuurd. En dan praten we echt over het 'aanschoppen'; voor versiebeheer en applicatiemodellering is integratie aanwezig met andere tools zoals Borland-Together of IBM-Rational, en voor softwaredistributie. Op deze manier worden de processen ondersteund, zodat de projectmanagers echt kunnen denken in increments, changes en software-releases.

Beide non-profiling centers hebben enige samenwerking met de anderen; dat gaat vooral via de CMDB waarin objecten zoals 'release' en 'programma-module' gedefinieerd staan. Ook de relatie tussen business-services en programma's kan worden bijgehouden. Voor onze focus, profiling, gaan we er niet dieper op in.

> Overal in deze twee centers zien we re-use: eenzelfde functionaliteit voor andere doelen ingezet

PROFILING-PRODUCTEN Wat opvalt binnen deze twee centers is het hergebruik. Heel veel producten zijn in de kern gebaseerd op 'session analysis', het fundament van LoadRunner. Deze techniek omvat:

a) Op netwerk-niveau 'sniffen' van applicatiecommunicatie, of het nu webgebaseerd is of client/server. Hiervoor worden meters ingezet in diverse netwerksegmenten.
b) Simuleren van het applicatiegedrag door naspelen van de bandopnames; voor één of voor honderden users.
c) Meten en analyseren van het communicatiegedrag onder deze nieuwe load-situatie, per betrokken server.

Er zijn dus in principe geen agents aanwezig binnen de applicatieserver of binnen de IDE. Maar er zijn wel twee aanvullende analysetechnieken: 'end user'-metingen (voor de overall responstijd, deels met agents) en 'mapping' om op basis van het sniffen de componentenarchitectuur te bepalen. En tot slot zien we in de 'monitor' modulen toch een stuk agent-achtige technieken opduiken, straks meer hierover.

We lopen nu eerst de producten van beide centers langs en de samenhang, en kunnen dan inzoomen op de profiling-kwaliteiten. Er zijn voor profiling twee kernproducten: LoadRunner en Diagnostics. LoadRunner is er primair voor de overall responstijd onder allerlei gesimuleerde omstandigheden. Diagnostics voegt daar breakdown-analyse voor specifieke omgevingen aan toe; en heeft daarvoor ook diepe kennis nodig van hoe

een .NET, of J2EE, of SAP of andere applicatie qua componenten en protocollen in elkaar steekt. Met deze twee namen hebben we ook de kern van het Performance Center gehad. Dat kent nog drie andere producten:

- *Tuning* is een op LoadRunner gebouwde methode voor performanceverbetering. Dezelfde packet analysis-tests worden gebruikt, doch nu niet om de knelpunten te vinden maar om stapsgewijs, na elke applicatie/configuratiewijziging, te analyseren waar verbetering te behalen is. Bijvoorbeeld door kennis van de vele configuratie-settings van een server, en vastleggen in workflow welke changes we hebben aangebracht.
- *Capacity planning* is de datawarehouse-broer van profiling; dit is een OEM-product van HyPerformix. Zowel testmetingen als productiemetingen worden in een database opgeslagen, en via trendanalyse wordt bepaald wanneer er overbelasting zal plaatsvinden en via welke sizing (zoals extra hardware) deze oplosbaar zijn.
- *Global Management* fungeert als globaal portal op LoadRunner *cum suis*, voor gedistribueerde consistente inzet.

Het Application Management gedeelte, oftewel BAC, is voor performance gebaseerd op dezelfde traffic analysis-metingen en op end user-bepalingen. BAC wil echter veel meer zijn dan alleen performance-analist, en duikt ook de wereld van systems management in. De positionering is vooral 'totaalparaplu' voor business services bovenop domein-meettools zoals Microsoft MOM of OpenView, maar ook 'performance-expert binnen productie'. Over de eerste rol kunnen wel wat twijfels geuit worden. Zo kan bijvoorbeeld de Mercury-console zelf geen correctiescripts lanceren, en is er geen netwerktopologie. Gezien de scope van dit artikel kijken we daar niet verder naar. Wij willen onze development-profiling kunnen relateren aan productieperformance en daarbij element measurements kunnen doen op elke betrokken server. BAC kan die taken aan. De modulen zijn, naast wederom Mercury Diagnostics:

- *Application Mapping*, voor het bepalen van de samenhang tussen applicatiecomponenten zoals web services.
- *End user monitoring*, als aanvulling op de packet metingen.
- *Service level management* en system availability management. Dit zijn echt 'business service paraplumodulen', die trekken op Diagnostics en end user metingen maar liever ook op echte server- en netwerkbeheertools zoals NetIq en NetView.
- *SiteScope*, een lichtgewicht serverbeheertool. Deze bepaalt beschikbaarheid en globale performance van vele soorten servers, en kan ook in de eigen subconsole correctiescripts lanceren.

Het meten gebeurt wederom niet met lokale agents maar met een ander soort metingen-op-afstand: remote beheer-API's. Zie verderop de bespreking van de Diagnostics 'monitors', deze code wordt gedeeld met Diagnostics en LoadRunner.

Verderop in dit artikel kijken we naar hoe de breakdowngegevens en element measurements verkregen worden. De nadruk op agentloze metingen moge duidelijk zijn, en die kennen in productiesituaties zo hun grenzen.

Performance Center modulen We lopen nogmaals langs de profiling-producten, doch nu alleen als ze direct verantwoordelijkheid dragen voor het invullen van onze wensen. En daarvoor gaan we soms nog iets dieper de productnamen in, en geven we een voorlopig oordeel over de sterke punten.

Het eerste product, LoadRunner, is zoals gezegd hoeksteen van het beleid. Diens capture-replay model hadden we al genoemd. Het instellen gebeurt vanaf de centrale console, de Loadrunner Controller. Het naspelen van sessies heet 'virtual users', en wordt vanuit testwerkstations gedaan. LoadRunner is bedoeld voor een veel breder scala dan alleen profiling; ook echte 'load tests' om de schaalbaarheid van een bepaald serverpark te meten zijn de doelgroep. Binnen ons wensenplaatje zorgt hij vooral voor de scenariobouw van profiling: de spreekwoordelijke 'minder RAM, verdubbelde load, of modemverbinding'. Want ook al draait de te testen applicatie prima in ons high-speed lab, eenmaal in productie kunnen andere spelregels gelden.

Een tweede kernproduct van Performance Center is Mercury Diagnostics. Dit baseert zich deels op het capture-replay model van LoadRunner maar voegt het nodige toe. Als we kijken naar één van de basisvarianten, Diagnostics for J2EE, dan kent die maar liefst drie 'capability's':

- *Transaction breakdown* oftewel triage, die precies aangeeft welke server-laag welke milliseconden opeet.
- *Application monitors*, die de J2EE-structuur en environment weergeven.
- *Deep diagnostics*, een agent-techniek die de meeste overhead met zich meebrengt en daarom vooral in de ontwikkelomgeving gebruikt zal worden.

Metingen Deze capability's corresponderen keurig met de drie soorten metingen die Diagnostics kent. De eerste soort heet 'protocol', en bestaat uit kennisverrijking van de capture-filters. Er zijn vijf hoofdgroepen in de protocolkennis:

- ERP/CRM, voor onder meer Oracle/PeopleSoft, Siebel en SAP.
- Web, waaronder HTTP, SOAP maar ook Citrix.

- Middleware: alle inter-server protocollen van J2EE en COM+ applicaties maar ook MQSeries en CORBA (de echte .NET-applicaties vallen reeds onder de Web-SOAP categorie).
- Databases: alle calls naar Oracle, SQL Server en DB2.
- Legacy: terminalverkeer voor DEC/Unix en IBM-mid-frames.

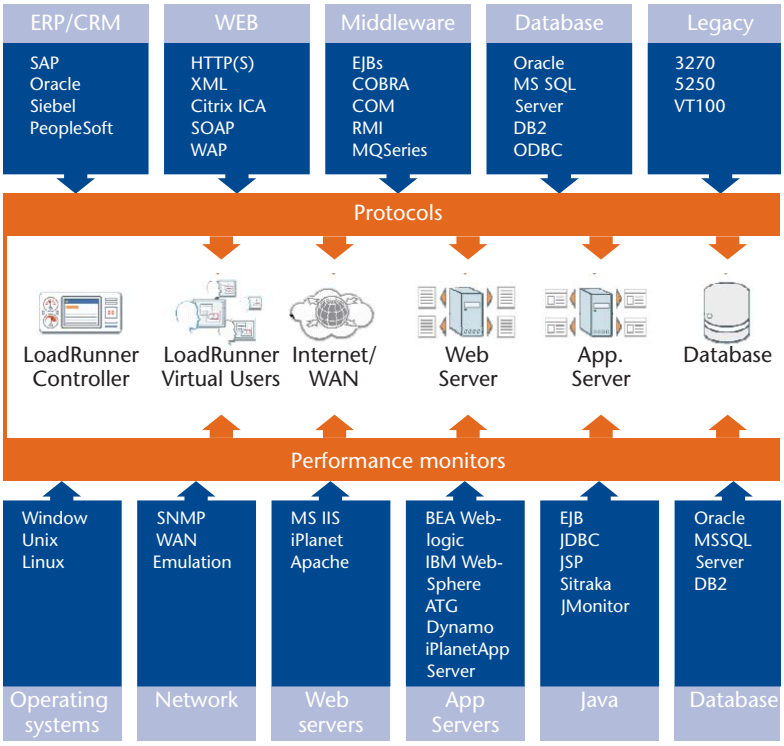
Deze kennis zorgt voor de breakdown-analyse maar komt ook weer terug bij de Application Mapping-module.

Het tweede metingentype heet 'monitor', en is enigszins on-Mercuriaans. Niet wat betreft de locatie waarop gedraaid wordt, want dat vindt nog altijd plaats op centrale meetmachines en niet lokaal per server. Maar wél qua vergaartechniek, want die is analoog aan wat we gewend zijn van pakweg OpenView of Microsoft MOM. Hij werkt met de gewone systems management API's van allerlei soorten servers en platforms, doch altijd via remote-aanroepen. Er zijn zelfs zes groepen monitoren:

- *Besturingssystemen*: Windows (WMI) en Unix/Linux (via remote het syslog bekijken en soms WEBM).
- *Netwerk*: via SNMP worden de 'zwarte dozen' bevraagd over hun gezondheid en andere meetgegevens, al gaat het niet direct om hun performance-metingen.
- *Webservers*: via de API's van onder meer Apache en MS-IIS.
- *Applicatieservers*: dit gaat via API's zoals JMX (J2EE) en weer WMI voor Microsoft.
- *Java*: hetzelfde laken een pak, via JMX worden nu gegevens binnen de applicatieserver getraceerd.
- *Databases:* dezelfde 'usual suspects' als bij protocollen, maar nu kijken we niet naar hun netwerkverkeer maar naar de specifieke beheer-API's.

Over de derde meetstijl, 'deep diagnostics' hadden we al kunnen vaststellen dat deze primair in de testomgeving gedraaid zal worden. Momenteel is er alleen naast tools voor Oracle-database en Siebel een J2EE-versie (voor onder meer WebSphere, WebLogic en Sun); aan een .NET wordt gewerkt. En deze meetstijl kent, een unicum binnen Mercury, een agent die echt fysiek aanwezig op elke te bewaken server; als een set (Java)-classes die samen een 'instrumentation plan' vormen. Hiermee wordt op vrij diepgaand niveau het servergedrag getraced, en dat trace-log wordt daarna door Diagnostics gecorreleerd met de andere twee parallel gedraaide agents voor een totaalbeeld. En daaruit blijken dan zaken zoals memory-leaks, deadlocks en EJB-data gerelateerde troebelen.

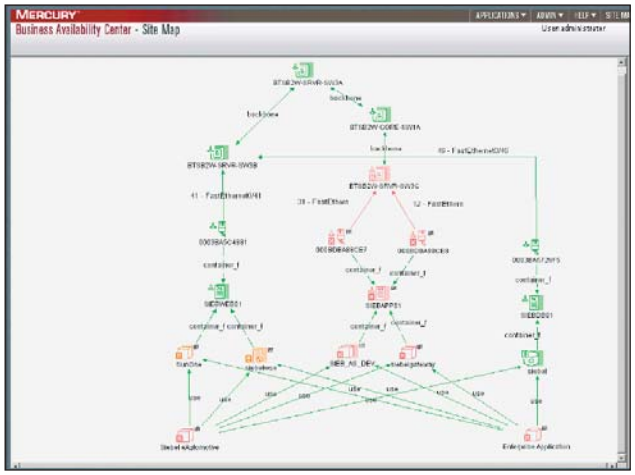
Business Availability Center We constateerden al bij onze wensenlijst: een eenmaal ontdekt performanceprobleem kan liggen aan slechte programmering of aan onjuiste machine-sizing. Om dit onderscheid te



Figuur 3. Lijst van protocollen en monitors.

maken hebben we element measurements nodig. De measurements bepalen de 'schuldige', samen met de sizing-richtlijnen van de middleware-leverancier en onze performance-breakdown. Als bijvoorbeeld onze netwerk-latency te hoog is, en uit de SNMP Monitor blijkt dat deze applicatie slechts tien procent van de bandbreedte toebedeeld krijgt, dan is dat een sizing-probleem en geen programmeerfout. Ook database-query's waarbij veel te veel van en naar disk geswapped wordt, duiden op sizing, terwijl query's die instorten omdat er gezocht wordt op een niet-geïndexeerd veld duidelijk aan de softwarebouwers liggen.

Voor productiemetingen kent Mercury het eerder geïntroduceerde BAC. Dat kent twee meet-modulen en een assistent. De eerste meetmodule omvat de protocollen en monitors die we al uit het Performance Center kennen. De protocollen worden nu vanzelfsprekend alleen gebruikt voor één virtuele user, en meten keurig de productiebreakdown, mits we op alle belangrijke segmenten meetmachines geplaatst hebben. De monitors meten het gedrag van alle relevante productiemachines, inclusief de netwerksegmenten (via SNMP-verkeer met de zwarte dozen). Deze meetmodulen kunnen we inzetten als deel van Diagnostics, maar ook eventueel als deel van SiteScope; of die tweede nodig is hangt af van de vraag of er andere systems management-tooling in productie aanwezig is voor onder meer het foutbeheer. Alle metingen kunnen historisch worden bewaard, en komen vervolgens terug in trend-analyses van het Application Management Dashboard. Hiermee



FIGUUR 4. Application Mapping resultaat.

is dus ook belastings- en performancegedrag over de tijd in te zien, en kunnen we bepaalde problemen voorspellen.

Triage: de overall-responstijd opgedeeld per component. Waar zit onze flessenhals?

Een tweede soort meetmodule omvat de 'end user monitoring' opties, veelal afkomstig uit de Topaz-bloedgroep. Er zijn drie opties:

1) *Real User Monitor*. Dit instrument zit op een netwerksegment waar er naar het productieverkeer 'geluisterd' wordt meestal via de spanning-port van een switch (dus zonder overhead te verzorgen op het netwerk zelf). Via packet-analyse wordt de responsetijd zoals eindgebruikers die ervaren voor vele users gemeten.
2) *Client Monitor*. Deze zit bijgeprikt op desktops, ofwel productie-users ofwel systemen waar synthetische transacties worden nagespeeld.
3) *Business Process Monitor*. Dit is echter optimaal re-use van elders ook al genoemde modulen: dit zijn dezelfde capture-replay opties die we ook als de 'protocols' van Mercury Diagnostics en LoadRunner kennen. En die geven, naast optioneel een breakdown per applicatiecomponent, ook altijd de overall-responstijd gemeten vanaf de meetmachine waar deze synthetische transacties gestart zijn.

De 'assistent' binnen BAC, tot slot, is Application Mapping. Die hadden we ook al eerder genoemd: hij bepaalt de samenhang tussen applicatiecomponenten. Omdat die ook de server-namen bevat, draaien we AM zowel in productie als in de test-omgeving. AM is ook erg nuttig voor het opbouwen van business service-overzichten in de Mercury SLM-module.

KRACHT EN ZWAKTES Nu we het profiling-portfolio twee maal langsgelopen hebben blijkt dat Mercury, ondanks het sterk leunen op de capture-replay techniek, op punten waar dat nodig is performance-kennis via andere technieken erbij haalt: via remote beheer-API's, via end-user metingen en een enkele keer (deep diagnostics) zelfs via fysieke agents. Daarmee scoort het grofweg gesproken even goed in onze profiling-wensenlijst als Compuware. Toch zijn er natuurlijk ook, net als bij de concurrenten, sterkere en zwakkere gebieden. Voor de volledigheid noemen we ook kort wat er minder goed is aan de aanpak.

Een eerste kanttekening is het vrijwel volledig leunen op remote beheer-API's. Nu kan er veel met standaards zoals WMI en JMX, maar lokale agents kunnen nog steeds méér - ook qua performance- en resourcemetingen. Ze hebben bijvoorbeeld toegang tot het lokale eventlog, en tot de volledige Windows PerfMon API. Dat stukje extra mist bij Mercury beheer, en kennen vele concurrenten wel. Een tweede, minder zware, kanttekening is het te elitair leunen op eigen netwerk-performancemetingen. Hoewel (in tegenstelling tot Compuware) wel toegang aanwezig is tot de statistieken van zwarte dozen via het SNMP-protocol, worden routers níet bevraagd op performancemetingen via RMON. In plaats daarvan leunt Mercury voor performance-breakdown volledig op de eigen sniffing-metingen op de testmachines, waarvan er heel wat geïnstalleerd moeten zijn op alle cruciale netwerksegmenten. Een derde 'beperking' zien we ook bij andere markt-spelers: nog iets minder support voor .NET dan voor J2EE. De 'deep diagnostics for .NET' module moet deze achterstand in een later stadium gaan inlopen.

CONCLUSIE Deze beperkingen nemen echter niet weg dat over het geheel genomen het volledige profiling-plaatje wordt ingevuld. Breakdown analyse in zowel test- als productiesituaties, element measurements in relatie tot de overall performance en trends. Daarmee blijkt de capture-replay techniek die oorspronkelijk voor load tests ontwikkeld werd, voor beduidend meer doelen inzetbaar te zijn. Wat ons betreft scoort Mercury prima als aanbieder van 'profiling optimization', en hebben ze dat deelproces ook redelijk geïntegreerd met een bredere change management- en systems management omgeving. Waarmee de claim van business process optimization in ieder geval voor een deel gehaald wordt. Chapeau, heren!

Erik de Ruijter RI is ICT-architect