

# Geek van de Week lost vooral problemen op

## PELI DE HALLEUX IS NOOIT KLAAR MET PEX

Robert de Ruiter

Om bij Microsoft tweemaal 'Geek van de Week' te worden moet je wel iets in je mars hebben. En gemeten aan het applaus aan het eind van zijn presentatie op SharePoint Connections 2010 klopt dat ook wel. Verschillende deelnemers komen hem na afloop een hart onder de riem steken. Kreten als 'cool', 'super' en 'geweldig' zijn niet van de lucht. Wie hij is? Peli de Halleux, 'Senior' Research Developer bij Microsoft.

Samen met collega Nikolai Tillman vormt hij de research-afdeling van Pex. Zelf is hij nogal nuchter over zijn werkzaamheden. "We lossen vooral problemen op", zegt hij. In de praktijk komt dit erop neer dat Nikolai en Peli bij Microsoft een radicale nieuwe benadering van unittesten ontwikkelden, die automatisch unittests genereert en bugfixes voorstelt. Dit leidt tot het product Pex, dat deel moet gaan uitmaken van de powertools in Visual Studio. In Visual Studio zijn gereedschappen ondergebracht voor het ontwikkelen van SharePoint-applicaties. Met Pex straks in de powertools kan een waardevolle bijdrage worden geleverd aan het testproces. De API van SharePoint dwingt je om integratietests uit te voeren. Dit betekent dat je altijd tegen een live SharePoint instance moet testen. Dit brengt veel kosten met zich mee op het gebied van performance en deployment.

### Isoleren

Isolatietechnologie maakt het mogelijk om in het geheugen te draaien. Dat is sneller. En daar bovenop genereert Pex automatisch tests op een manier, die vergelijkbaar is met hoe ontwikkelaars stap-voor-stap een applicatie ontwikkelen in de debugger. Omdat Pex de data- en de controlestroom begrijpt geeft het interessante input aan de code. Het komt er in het kort op neer dat Pex een goede assistent is om die bugs te vinden, die je met integratietests over het hoofd zou zien.

Dit past goed binnen de verbeteringen die in Visual Studio zijn aangebracht om de SharePoint-ontwikkelaar tegemoet te komen. In Visual Studio is speciaal aandacht geschonken aan SharePoint en de researchers wij hebben geprobeerd aan de testkant wat extra waarde toe te voegen. Veel ontwikkelaars vinden het prettig om ook unit tests uit te voeren op de ontwikkelde code.

Pex staat nu nog in de kinderschoenen, geeft Peli toe. De bedoeling is echter dat het uiteindelijk als powertool wordt toegevoegd aan Visual Studio. Op dit moment worden delen van Pex vrijgegeven om feedback te krijgen en het product te vervolmaken. Peli kwam in 2006 bij het Pex-project, dat anderhalf jaar eerder door Nikolai Tillman werd opgezet. Daarvoor werkte Peli deze aan Spec Explorer. Pex is eigenlijk een afgeleide van verschillende eerder door de afdeling research ontwikkelde testtools. Allemaal gebaseerd op de gedachte dat deze tools ervoor zorgen dat de software doet waarvoor deze is ontwikkeld. Tillman en De Halleux werken nu binnen het Research and Development Department van Microsoft samen aan de verdere ontwikkeling van het unittest-instrument. "We zijn dus een research-unit van twee man binnen dat grote Microsoft", stelt Peli vast met een brede grijns. "Onze voorgangers hebben ook geprobeerd modellen te schrijven, maar dat is lastig. Je moet een expert zijn om een model te schrijven. Begrijpen wat een

model doet. Uiteindelijk kwamen we uit bij Pex: een service met parameters. In feite is dit ook een model; een algebraïsche specificatie".

Pex werkt met stubs en moles. Stubs zorgen voor de interfaces. Hierdoor kan een module geïsoleerd worden getest van andere modules. Als je goed hebt gecodeerd zijn de verschillende componenten van elkaar gescheiden met interfaces. Stubs gebruiken die interfaces. Stubs zijn We hebben dit bewust gescheiden van de moles, vanwege de implicaties. Als je moles gebruikt, moet je er random instrumentation op loslaten. Dat is een veel zwaardere procedure dan het implementeren van de interfaces, wat stubs doet.

### Beavers?

"We gebruiken in Pex ook de naam stubs niet meer. Niemand kon dit vinden, dus we hebben alles gerenamed naar moles. Dat konden we in dit stadium van incubation-project nog doen. "Moles verkopen beter, ook al wil je ze liever niet in je tuin", zegt Peli lachend. "We hadden al de vis 'Pex' en hiervoor zochten we een andere dieren naam. We hadden voor de behaviour-typen eerder ook al de 'Beavers' bedacht. Tot we erachter kwamen dat die in het engels van het Verenigd Koninkrijk ook een geheel andere betekenis hadden..... Toen zijn het 'B-types' of 'Behaviours' geworden".

"Het idee erachter is: heb de voordelen van integration-testing zonder de bekende na-

## Rare naam...

Zo bijzonder als zijn werkzaamheden en de resultaten daarvan is ook zijn naam. Bij zijn geboorte in het Waalse deel van België werd hij Jonathan de Halleux gedoopt. Het idee kwam van het bekende muziekstuk van Neil Diamond 'Jonathan Livingstone Seagull'. Dat gaat over een zee-meeuw, die zich van alle andere meeuwen onderscheidt omdat hij van vliegen een kunst maakt. Al na twee maanden gaan zijn ouders Jonathan Peli noemen, dat is afgeleid van Pelikaan. Niemand weet zich meer te herinneren waar dat vandaan kwam. Dat 'pelikaan' Jonathan, die eigenlijk dus zee-meeuw is, zich onderscheidt op zijn vakgebied is wel duidelijk. Al enkele jaren nadat hij zijn studie toegepaste wiskunde aan de Katholieke Universiteit van Leuven afrondde viel het oog van Microsoft op hem en werd hij uitgenodigd om bij de Software Engineering Group in Redmond te komen werken.

delen. Het voordeel van integration-testing is dat het makkelijk is. Je zet de final state van je systeem. Het enige nadeel is dat het traag is. Het doet er niet toe hoe je er komt, omdat het niet het juiste abstractie-level is. Zo werkt dat met mocks. Prima, maar je betaalt er een zware prijs voor wanneer je later iets aan de software wijzigt. Dan moet je de tests ook opnieuw schrijven. Wij denken dat het werken met

product. Het is op veel serieuze code losgelaten en heeft inmiddels zijn nut bewezen. En je zou kunnen stellen: iedere tool is een tool. Waar komt de waarde dan uit? "Uit het feit dat je als ontwikkelaar sneller kunt ontwikkelen, bijvoorbeeld. Als je kijkt naar Test Driven Development: daar heb je hele korte iteraties. Onder de seconde. Maar Pex is ook een heel interessante tool. Stel dat een consultant 150

ties. Dat is voor ons het beleg op het brood. Neem Dell als voorbeeld. Die hebben veel inspanning gestoken in het ontwikkelen van betere software, meer getest, betere code. Het gevolg is dat ze veel minder systeemstoringen hebben en dat de performance sterk is verbeterd".

Peli noemt Pex zelf 'a pretty good tool'. Het gebruikt veel geavanceerde technologie, veel G3, dat ook door Microsoft is ontwikkeld. Er zit veel research in. Het is een van de beste unittests die nu beschikbaar is. Toch is hij nog niet helemaal tevreden.

"We hebben nog geen tijd gehad om er een beter distributiemechanisme in te bouwen. Meestal heb je een fixed testsuite, die je iedere dag op je build loslaat. Terwijl je product verandert, verandert je unit test suite niet. Je betaalt er eenmaal voor en daar moet je het dus mee doen. We hopen dat we met Pex kunnen bereiken dat telkens wanneer je iets aan je code verandert, ook de testsuite daarop reageert".

"Een andere functionaliteit betreft regular expressions. We analyseren die, maar hebben er geen speciale controle voor. We kunnen niet echt XML-files genereren. Er zijn dus best nog uitdagingen voor Pex. Voor de professionals zijn de behaviours ook nog niet echt klaar. Nog niet echt gevalideerd tegen volledige applicaties. So wieso zijn er nog heel veel uitdagingen. Op het gebied van scalability bijvoorbeeld. We hebben nog een volgeschreven whiteboard met onderwerpen, die we nog willen aanpakken. Waarschijnlijk zullen we nooit echt kunnen zeggen "nu is het klaar". Bij Research zijn we voortdurend op alle fronten actief. We proberen een goed image te kweken in de academische wereld, maar we willen ook steeds betere ondersteuning bieden voor de ontwikkelaars. Kortom: het is geweldig om in de testsector te werken. Waarschijnlijk raken we hier ook nooit zonder werk".

## Referenties

Peli's Farm: <http://blog.dotnetwiki.org/>  
Zie ook .NET Magazine #1 van 2009 voor een technisch artikel over Pex door Alexander Nowak.



PELI DE HALLEUX GAF TIJDENS SHAREPOINT CONNECTIONS EEN BOEIENDE SESSIE OVER HET GEBRUIK VAN MOLES IN PEX.

behaviours een betere oplossing is. Maar dat moeten we nog valideren. Het is nu eenmaal research waar we mee bezig zijn. We laten het eerst nog door heel veel SharePoint-ontwikkelaars proberen". Versie 0.21 van Pex is net klaar. Hoe veel werk er nog moet worden verzet voor het programma productieklaar is, is in feite aan de klant. Als die er voldoende functionaliteit in ziet kan Pex de markt op. Wat Peli betreft is Pex nu al een zeer bruikbaar

euro per uur kost. Je kunt Pex in een minuut deployen. Dan heb je nog 20 seconden nodig voor IIS. Tel maar uit je winst", zegt De Halleux.

## Filosofie

En hij vervolgt: "Het is ook afhankelijk van de vraag hoe veel je wilt investeren in kwaliteit. Wat is de filosofie van je bedrijf? Ben je een stel cowboys of streef je net als wij naar correcte software met goede specifica-

.....  
**Robert de Ruiter**, is hoofdredacteur van .NET Magazine

