

Zeven jaar geleden, in SRM 5/2001, presenteerde Software Release Magazine NetPresenter. Toen, net na het uiteenspatten van de internetzeepbel één van de weinige kleine bedrijven die zich met een nieuw internetgerelateerd product staande wisten te houden. Tijd voor een hernieuwde kennismaking om te zien of mens en technologie hetzelfde gebleven zijn.

NetPresenter

Revisited

Bij aankomst in Beek blijkt NetPresenter nog steeds in dezelfde oude slagerij gehuisvest te zijn. De garagementaliteit is er kennelijk nog steeds, ondanks de Fortune 500-klanten.

Zeven jaar geleden verkocht NetPresenter een product waarmee het heel gemakkelijk was om binnen netwerken boodschappen te verzenden die desgewenst op het scherm van de gebruikers opdoken. Het werkte met een heel kleine engine, die meen ik gebaseerd was op Amiga-software. Is die engine er nog?

Hoën: 'De filosofie van NetPresenter is nog steeds een compacte engine of player die gemakkelijk organisatiebreed uit te rollen moet zijn. Wij hebben volledige controle. Onze client is nog steeds een media engine van 350k. Dat is belachelijk weinig en bovendien werkt hij zonder dependencies. De afhankelijkheid van libraries of plug-ins van derden maakt het organisatiebreed verspreiden onmogelijk. De systeembeheerder heeft vaak mixed environments, met Windows 2000, of zelfs NT 4.0. Wij werken met een client zonder afhankelijkheden en dezelfde client wordt gebruikt voor de desktop. Daar werkt hij als screensaver, een pop-up die echter niet door de pop-upkiller wordt gehinderd, omdat hij buiten de browser om gaat. Aan de andere kant wordt hij gebruikt voor narrowcasting, dus voor de grote schermen. Feitelijk is het moeilijker om dit soort clients te laten draaien in een corporate environment met werk-

plekken die continu worden gebruikt, dan dat je een standalone-kistje hebt waarop je kunt draaien wat je wilt. Dat stelt technisch niets voor; het probleem begint bij de beheersbaarheid, schaalbaarheid en robuustheid. We hebben applicaties in webbrowsers die continu draaien, 24 uur per dag. We constateren tot op de



dag van vandaag kleinere *memory leaks*, meestal van third party-applicaties. De geheugenbescherming van .Net doet wel zijn werk, maar er wordt ook nog veel unmanaged code gebruikt. Microsoft kan nog zo'n goede code schrijven, op het moment dat een andere applicatie die geheugenruimte niet teruggeeft heb je toch een probleem. Iemand die zijn browser opstart en er de hele dag mee werkt merkt er weinig van. Mensen die een browser gebruiken in een kiosk-toepassing moeten speciale voorzieningen toepassen om het geheugen te recupereren. Ook nog onder .Net, alhoewel het probleem daar wel minder is geworden. Kortom: die compacte standalone-client bestaat nog steeds. Die vind je dan ook bij kerncentrales op iedere pc, op vliegtuigschepen, bij DSM op 22.000 werkplekken. Ook met een LAN-verbinding naar Maleisië en dat zijn geen glasvezelverbindingen, dus je moet er toch voor zorgen dat ook het LAN niet plat gaat. We hebben het concept uitgebreid, er security aan toegevoegd, scheduling en een reeks nieuwe features. Wat ook nieuw is: het koppelen aan bestaande content management-systemen, legacy-systemen. Met name RSS-XML leent zich daar perfect voor. Dan hoeft je als het ware maar achterover te leunen en de content wordt automatisch gevisualiseerd. Dat pushachtige concept van NetPresenter is dus nog steeds realiteit. Het gekke is nu juist dat het door de *information overload* steeds actueler wordt. Mensen gaan steeds minder goed hun mail lezen, het communiceren van de essentie is weer helemaal hot. Hoe krijg je de koppen binnen de organisatie? Hoe communiceer je dat virusalarm, of kondig je aan dat je onderhoud gaat verrichten aan de werkplek? Of dat die nieuwe sectie of dat nieuwe systeem op intranet beschikbaar is? Het visueel actief tonen van content is eigenlijk groter dan ooit.

Ondanks de moderniteit van het oude concept, volgt NetPresenter ook de nieuwste technische ontwikkelingen.

Hoën: "Aan de ene kant hebben we koppelingen met CMS-sen, aan de andere kant hebben we RIA's. Echte eigen contentapplicaties waarmee mensen dus functionaliteit die men normaal in een clientapplicatie aantroef, in hun browser krijgen. Men kan dus eigenlijk eenvoudig berichten toevoegen in een webinterface.



Wij noemen dat onze messageserver. Daar kun je berichten aanmaken, slides maken, plannen, targetten, kleurenplaatjes en video's gebruiken, allemaal dingen die in een klassieke client/server-omgeving zaten. Met RIA's kun je dus ook heel gemakkelijk dienstverlening realiseren over het open internet en hoeft je je niet te beperken tot dienstverlening binnen organisaties. Je bijvoorbeeld een makelaar een clientapplicatie geven over het open internet."

Zeven jaar geleden werd er veel in C geschreven, is dat nu .Net?

De meeste van onze serverapplicaties zijn nu op basis van .Net geschreven. De client nog niet, daar wachten we waarschijnlijk mee totdat Vista mainstream geworden is. Het probleem buiten Vista is dat je niet weet welke versie van .Net framework-organisaties geïnstalleerd hebben.

Maar de huidige client draait wel op Vista?

Ja, als je je code netjes schrijft dan is dat geen enkel probleem. Clean code die van Windows95 tot Vista loopt is voor een applicatie redelijk uniek. Dat lukt omdat we het hele traject in eigen beheer houden.

Welke talen worden er nu gebruikt?

We komen van C++ en daarom is C# een logische stap. PHP wordt nog wel ingezet, maar op lange termijn wordt dat ook allemaal C# en ASP.Net. Microsoft gaat hier gewoon 100 % voor en zodoende durven wij die keuze ook te maken. Wij

werken nu al met Silverlight. Onze verwachting is dat de acceptatie in de corporate wereld vrij algemeen zal zijn. Buiten de corporate wereld is het absoluut Flash, maar daarbinnen toch meer Silverlight. Alhoewel de grafische performance en featureset van Silverlight wel nog een lange weg moet gaan. De 'grafische versnelling', door het gebruik van de GPU (processor van de grafische kaart), is anno 2008 nog steeds geen realiteit. Een tickertape in WPF, als die al mogelijk is in Silverlight – een subset van WPF – loopt nog even langzaam als in Flash. We houden graag vast aan één taal, of een framework in dit geval. Minimaal 95% van onze potentiële klanten gebruikt Windows, dus er was ook geen noodzaak om op Java over te gaan. Wij werken met Silverlight van prototype naar prototype. Soms zie je dat bepaalde functionaliteit nog ontbreekt, maar die wordt dan al wel aangekondigd met Silverlight. .Net is als framework stevig genoeg om je toekomst op te baseren. We merken nu al dat de turnaround time development cycles stukken korter zijn dan met traditionele softwareontwikkeling in C++. We hebben ook PHP-applicaties die deels uitgefaseerd worden en deels naar ASP. Net en C# gemigreerd worden, dat is altijd een afweging. Vroeger deed je alles in PHP en C, nu heb je een aantal modules op basis van .Net. Als je bepaalde functionaliteit overzet, dan merkt de gebruiker daar eigenlijk niet zo veel van. Ondertussen ben je het fundament wel aan het vernieuwen. Het zijn lange trajecten,

voordat wij die software vrijgeven. We hebben een miljoen van die clients verkocht. Als NetPresenter plat gaat, dan hebben we een probleem. Het zijn vaak levensreddende systemen. We zijn nu naar aanleiding van een schietincident bezig Amerikaanse scholen te voorzien van een systeem waarmee ze heel snel, via ieder pc-scherm kunnen waarschuwen. Dan moet je er voor zorgen dat het goed functioneert, anders heb je zo een rechtszaak aan je broek. Ondanks de vernieuwingen is die client gewoon rockstable. We houden ook alles in eigen hand, en dat scheelt.

Zeven jaar geleden werd veel software-ontwikkeling in de V.S. gedaan, in feite was er sprake van offshore.

Hoen: "We hebben alles hiernaartoe gehaald. Voor routinewerkzaamheden is offshore geen probleem, maar met name als je innovatief moet zijn, dan is het heel moeilijk. Je bent heel veel overhead kwijt aan communicatie. Mensen komen vaak net van de universiteit en missen een referentiekader. We hebben een zestal programmeurs hier in Beek. Wij merken heel duidelijk de kruisbestuiving. De informele communicatie van programmeurs bijvoorbeeld tijdens de lunch is eigenlijk essentieel. Je mist heel veel als je die niet hebt."

Voor broadcasting op grote schermen wordt deels dezelfde programmatuur gebruikt als voor die op PC's (Foto: NetPresenter)



Dan moet je echt exploratief programmeren, dus programmeren in prototypes

Is er nu nog veel over van de software uit de tijd van het vorige interview?

Hoen: "Als je dat nu opnieuw zou moeten programmeren, dat zou niet te doen zijn. Er zitten zo veel manjaren in, met name op het gebied van stabiliteit, breedte en caching. Het is gewoon handwerk in C++, geen .Net libraries, maar het staat wel als een huis. Soms kun je na een aantal jaar met een library call wel twee maanden werk vervangen. Enerzijds vinden de programmeurs dat fantastisch, anderzijds denken ze, goh... Met name zoiets als WCF neemt toch visueel een hoop complicaties weg. Binnen het grote .Net-framework ontdekken we nog iedere dag opnieuw dingen."

Dat framework is ook nauwelijks meer te overzien, zeker niet voor één ontwikkelaar.

Hoen: "Ja, het is veel. Vroeger kon je overal van op de hoogte zijn, en tegenwoordig kun je alleen maar een specialist zijn binnen een deelgebied. Soms moet je ook niet te ambitieus willen zijn. Ontwikkelaars willen dan alles omzetten, waarbij ik ze dan een beetje moet afrem-

men: rustig aan, doe het nu per module. Je zult er een strategie voor moeten hebben: waar heb je nu het meeste voordeel van als we het in het nieuwe platform doen. Dat is een heel zoektocht, met trainingen en cursussen en noem maar op. Er volgt altijd wel iemand een cursus. Het lijkt ook wel alsof het accelereert. Tegenwoordig moet je heel veel keuzes maken: ga je om, wat doen we wel, wat doen we niet. Het probleem is dat bepaalde dingen vaak zo nieuw zijn dat Microsoft ook nog niet weet wat er precies mogelijk is. Dan moet je echt exploratief programmeren, dus programmeren in prototypes: 'dit zou theoretisch moeten kunnen, maar kan dát eigenlijk wel?' Dan pak je dus dat idee op, reserveer je er tijd voor en zeg je: probeer maar. Dan kijk je of het geheugen lekt, of de applicatie stabiel is, hoe de performance is en dergelijke. We weten wel dat Microsoft dat soort dingen verfijnt, maar exploratief programmeren, zeker als je op een bepaald concept je hele kaartenhuis wilt bouwen, is noodzakelijk. Dat is enerzijds het verkennen van de architectuur en de mogelijkheden ervan en anderzijds is het een leerproject voor de programmeurs. Het is perfect om door middel van fijnomlijnde projectjes een framework te leren kennen, zodat je niet helemaal nat gaat op het moment dat je op client/server-architectuur ook weer verder gaat."

Is er een bepaald methodologie, wordt er agile gewerkt, zijn er daily builds?

We maken wel daily builds en er is wel één programmeur die de toon aangeeft, maar er is geen unified approach. Enerzijds zijn er de mensen die low level PHP doen en de migratietrajecten, en anderzijds zijn er de oude C++ - programmeurs en die komen toch uit een andere richting. Wij waren vroeger wel heel erg ad hoc bezig. Het is natuurlijk een open deur, maar het goed naar specificaties vertalen van een feature is fundamenteel.

In het verleden deden wij dat niet en dan waren we programmerend aan het prototypen. Dat is verschrikkelijk, terwijl je die simulatie op papier gewoon kunt doen. Je kunt dan observeren hoe een gebruiker klikt en aan de hand daarvan ga je gewoon je prototype aanpassen. Als je test op papier en je ziet die C++-programmeur dezelfde fouten maken als een computerleek, dan weet je dat je een probleem hebt. Het .Net-framework, met name bijvoorbeeld XAML, maakt het wel een stuk eenvoudiger dan vroeger. Die grafische weergave kun je voor een belangrijk deel scheiden van wat eronder ligt, dat geeft ons wel wat meer flexibiliteit. Aan de andere kant is XAML wel een beetje een

brei, het is niet zo gemakkelijk met de hand te bewerken als HTML.”

Speelt de gebruiker ook verder nog een belangrijke rol in het ontwikkelproces?

Hoen: “De gebruiker weet niet altijd wat je wilt, dat is wel een beperkende factor. Maar voor het beoordelen van de usability is hij onontbeerlijk. Blijven kijken naar de gebruikers, ook tijdens trainingen is de opdracht: blijf observeren. Als je de user interface niet aanpast, dan krijg je het trouwens toch op je brood, want dan heb je hogere supportkosten..

Hoeveel mensen telt NetPresenter?

Hoen: “We hebben nu twaalf man vast in

diens. We hebben wereldwijd resellers met kantoren in onder meer de V.S, London en Duitsland. Dat zijn zo’n dertig man over de hele wereld; partners die niet op de loonlijst staan maar wel full time met NetPresenter bezig zijn op commissiebasis. Sony bijvoorbeeld is deels geïnitieerd door een partner. Grote bedrijven in Nederland doen we meestal direct, maar we hebben ook mensen in het Midden-Oosten zitten. Aan olieraffinaderijen wordt wel via lokale partners verkocht. Tijdens de tweede Golfoorlog is er een raket op Koeweit-stad gevallen, op het meest prestigieuze winkelcentrum daar. Dat maakte de noodzaak duidelijk de eigen mensen te bereiken in dat soort

Uit SRM 5/2002

NetPresenter: Groot geworden door klein te blijven

Oorspronkelijke Amiga-software draait nu op 350.000 werkplekken.

NetPresenter presenteerde zich onlangs in een persbericht als een ‘Nederlandse dot com survivor’. Het blijkt een klein bedrijfje te zijn, dat echter menige multinational onder haar klanten telt. Uit onderstaand gesprek blijkt, dat het bedrijf groot geworden is, door klein te blijven, hoe cryptisch dat ook klinkt. Dat geldt zowel voor financiële principes van het bedrijf als voor het resourceverbruik en de netbelasting van de producten.

[...]

De basis voor het succes van NetPresenter werd al gelegd in Hoen’s jeugd. Hij was een wizzkid, een eerste generatie hacker zoals hij het zelf noemt. Hij begon met Sinclair ZX80 (‘had je een tekstverwerker nodig dan schreef je hem zelf, desnoods in Assembler’), stapte al solderend en uitbreidingen bouwend over op het Amiga-platform en begon, toen hij economiestudent was, zelf chips te verhandelen.

[...]

De Amiga was een van de eerste echte multitasking platforms voor de consument. Hoen ontmoette een Amerikaan, een van de auteurs die schreven voor de Amiga in zuivere C-code. De Amerikaanse programmeur had daar toen al een auteursomgeving voor geschreven lijkend op Director. Hij was al in 1985 begonnen een hypertext-taal en scripttaal te ontwikkelen.

[...]Het faillissement van Commodore noodzaakte Hoen toch andere dingen te gaan doen.

Hoen: ‘Amiga was heel sterk in kabelkranten. Wij hadden al heel vroeg stereogeluid, animaties en duizenden kleuren op een scherm. De kabelkranten liepen allemaal op basis van de Amiga. Wij verkochten die kranten bijvoorbeeld aan Rijkswaterstaat, die er tijdens de overstromingsramp in januari 1995 – wij noemen dat als Limburgers overigens de watersnoodramp – veel gebruik van heeft gemaakt. Wij hebben toen een noodkabelkrant gemaakt met waterstanden en evacuatieplannen. Dat werkte fantastisch en daaruit bleek dat er een grote behoefte was aan snelle communicatie.

[...]De problemen zaten in de nieuwe 32-bits compiler, waar kennelijk fouten in zaten. Daarmee verloren we een paar maanden. Maar ondanks dat hebben we medio 1995 die code geport hebben naar Borland C, en hadden we versie 1.0 van NetPresenter.

[...]De kern van de programmeerafdeling van NetPresenter wordt nog steeds gevormd door de inmiddels brood met beleg etende Amerikanen. Eigenlijk is NetPresenter een voorbeeld van offshore ontwikkeling.

[...]Die auteursomgeving kregen wij gewoon voor NetPresenter cadeau – een hele hypertext-taal. Dat scheelde ons tien manjaren programmeren. Zo’n script engine met interactieve elementen, met een wipe-engine waarbij effecten over het beeldscherm kunnen lopen. Het is wel elegant dat we wat het script betreft gekozen hebben voor

hypertext, want dat sluit nu perfect aan bij huidige standaards als HTML. Als je het vergelijkt met PowerPoint: dat is gewoon een executable, daar kun je niets aan manipuleren. Met een scripttaal kan dat wel. Het manipuleren van een script maakt allerlei interessante oplossingen mogelijk.’

[...]Van de Amiga-filosofie erfde NetPresenter het idee van “thin client, low bandwidth”. De kleine langzame processor en het naar huidige maatstaven verwaarloosbare geheugen, dwong de programmeurs kleine compacte code te schrijven. Hoen: ‘De player van NetPresenter is al jarenlang 250k, dat is de hele engine. Een kleine, compacte en robuuste player, die gewoon het script eet. Gaat er iets mis, dan slaat hij dat gewoon over, want de player moet gewoon altijd overeind blijven. De uitdaging was, om op overvolle netwerken en oude pc’s goed te blijven functioneren, zonder dat de organisatie later zou kunnen zeggen, dat we het netwerk omlaag brachten.

[...]Think big, act small, die garagementaliteit. We hebben een oude slagerij gekocht, verbouwd voor een bescheiden bedrag, en van daaruit zijn we gewoon blijven werken. Het was echt ieder dubbeltje omdraaien.’

[...]Het concept is bewezen, het werkt. We hebben onze marktniche gevonden en bouwen die geleidelijk uit.’

Laat daar geen misverstand over bestaan: **corporate sales** zijn gewoon heel moeilijk

situaties, want een sirene werkt dan niet. Nu kunnen ze alarm geven via de pc, dus op de werkplek en via schermen. Wij hadden niet verwacht dat we onze systemen in Koeweit zouden verkopen, ook niet in India. Maar je moet niet de illusie hebben dat je daar met dezelfde regels en voor dezelfde prijzen kunt verkopen als in het Westen. Je ziet ook een heel duidelijk verschil tussen de Europese prijzen en die in de V.S. Software is in Amerika vele meer commodity dan hier. Je kunt geen global pricing hanteren, want er zijn altijd verschillen. Maar het werken met locale partners op commissiebasis gaat erg goed. Dat is gewoon een wat veiliger structuur voor een kleine softwareonderneming om mee te beginnen; de baten gaan voor de kosten uit.”

NetPresenter heeft als kleine onderneming zeer grote ondernemingen als klant. Het zal niet gemakkelijk geweest zijn die over te halen.

Hoën: “Als je tegenwoordig kijkt waar de meest gekke dingen gebeuren, dan is dat bij grote organisaties. Overnames, venture capital, equity funds. Het gekke is dat de kleine werkgevers veelal betrouwbaarder zijn dan de grote werkgevers. De programmeur/lezers kan ik dan ook aanraden vooral voor de kleine bedrijven te gaan werken. Natuurlijk heb je wel wat meer te bewijzen als kleine marktpartij en je zult er voor moeten zorgen dat je tot een bepaald marktaandeel komt. Heb je dat gerealiseerd en heb je de accounts, dan is er ook geen discussie meer. Ik heb wel het idee dat er in Amerika minder discussie is dan in Europa. Maar het kost wel tijd. Wij hebben tegen de stroom in moeten roeien. Midden jaren negentig kwam bijvoorbeeld Lotus Notes op en daarvan werd letterlijk gezegd: hiermee gaan we al onze communicatieproblemen oplossen. Zo wordt dat dan verkocht binnen grote organisaties, nog steeds. Ik laat klanten mijn product gewoon proberen

én ik maak van mijn succesvolle klanten een podium. Wanneer je als leverancier zegt dat iets heel gemakkelijk is, dan maakt dat toch minder indruk dan wanneer een klant dat zegt. Met een bericht op de voorpagina van FD dat we Sony on product ging gebruiken hebben we weer een paar ander Fortune 500 klanten binnengehaald.

Maar laat daar geen misverstand over bestaan: corporate sales zijn gewoon heel moeilijk. Dat zit hem niet in het product, daar is geen discussie over, maar het zit hem in het complexe beslissingsproces binnen grote organisaties. Sommige grote banken moet je niet zien als ondernemingen, die kun je beter zien als ministeries. Dat zeggen ze zelf ook, dat is gewoon heel ambtelijk. Een voorbeeld: bij het uitrollen van een project bij een Nederlandse bank, moet er een projectmanager zijn. Die projectmanager kunnen ze pas krijgen als er budget gealloceerd is, en dat budget krijg je pas gealloceerd als er een projectmanager is! Het zijn de ambtelijke structuren die het uitrollen soms jaren kunnen vertragen. Als er een noodzaak is, gaat het wel vaak snel. Sommige organisaties zijn echt snel. Bij Sony bijvoorbeeld was het snel uitgerold in heel Europa. Maar bij de meeste grote organisaties zijn zo veel partijen bij een project betrokken dat je echt moet uitkijken dat het niet te lang duurt.”

Het gebruiken van klanten als podium werkt ook goed binnen de organisatie:

Hoën: “Het gaat uiteindelijk om de toepassing. De wetenschap dat met onze software boeven gevangen zijn en dat er een vermist kindje is teruggevonden doet mensen wel iets. Ontwikkelaars zitten toch vaak in een hokje en die komen er niet meer uit. Wij brengen heel bewust die mensen in contact met klanten. Wanneer een klant een ontwikkelaar aanstoot en zegt: ‘die software van jou, geweldig!’, dan werkt dat. Die program-

meur ziet dat niet, die is met structuur en modulen bezig, maar op het moment dat hij denkt ‘hè, die mensen vinden wat ik geschreven heb geweldig’, dan is dat soms beter dan loonsverhoging! Het werkt, dat stukje erkenning, zowel op het niveau van de programmeur als van de totale organisatie. Wij zijn een trotse organisatie, dit is iets waar we in geloven. Het is misschien een naïeve illusie, maar we denken dat we de wereld toch een heel klein beetje beter maken. En we zorgen er ook nog voor dat onze kinderen te eten hebben. In ons geval hebben we gigantisch voor de muziek uit gelopen: in 1996 kregen we een innovatieaward van Byte, omdat we een van de eerste push-bedrijven ter wereld waren. Het heeft bij wijze van spreken tien jaar geduurd voordat de rest van de wereld daar mainstream achter ging staan. Dat moet je durven: ‘Sometimes succes is sticking around long enough’.”