

# ‘Klanten vragen om proven technology’

## Bevolkingsonderzoek borstkanker is zeer centraal opgezet

*Grote projecten worden niet altijd binnengehaald door grote bedrijven. In het geval van het Landelijk Bevolkingsonderzoek Borstkanker was het RADventure uit Maarsse - een bedrijf dat ontstaan is uit een samenvoeging van Advantage Software, de distributeur van Clarion producten in de Benelux en Extensie, een softwarehouse. Men voert fixed price, fixed time en fixed quality-projecten uit met Clarion. Een gesprek met ing. Erik Pepping, projectleider en algemeen directeur van RADventure, over dit project, PL/SQL versus Java en meer.*

In oktober 1999 is RADventure gestart met het ontwerpen. Er is vier maanden ontworpen, waarvan twee maanden aan het basisontwerp gewerkt is en twee maanden aan het detailontwerp. Daarna heeft er een bouwperiode plaatsgevonden van zeven maanden. Full time werkten er twee en halve personen in deze periode op het systeem. De eerste stichting is 1 januari 2000 live met het systeem begonnen, na test en trainingsperiodes. De laatste twee stichtingen gaan pas dit jaar over.

Hoe lang heeft het precies geduurd, voordat het project echt in de lucht was?

Pepping: ‘We hebben ervoor gezorgd dat de mate van gedetailleerdheid in het detailontwerp niet te groot werd. De mensen in het werkproces staan zwaar onder druk omdat er behoorlijk is bezuinigd. Dat betekent dat ze niet veel tijd hadden om de zaak op zijn merites te beoordelen. We hebben een prototype gebouwd om te kunnen tonen hoe het er ongeveer uit zou gaan zien. Dat hebben we gepresenteerd en dat heeft wel heel erg geholpen.’

### Planningsproces

Het project is in de huidige opzet alleen mogelijk vanwege het feit dat Nederland - als een van de weinige landen in de wereld - beschikt over een GBA, een gemeentelijke basisadministratie. Pepping: ‘Elke Nederlander die in Nederland staat ingeschreven,



Ing. Erik Pepping, projectleider en algemeen directeur van RADventure

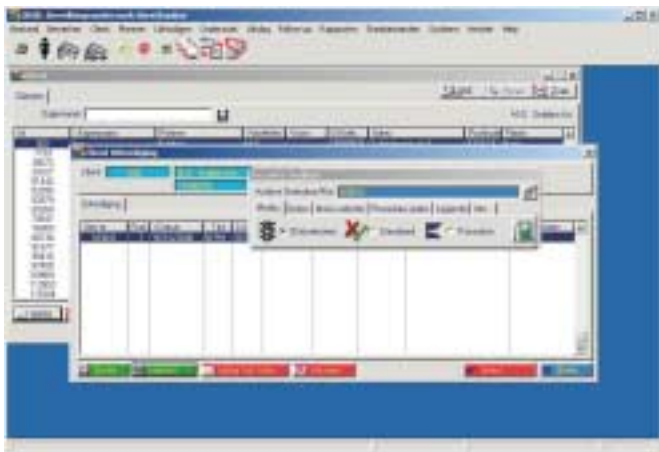
is geregistreerd in de GBA. Na de koppeling van het GBA wordt een vrij complex planningsproces opgezet. Want om ervoor te zorgen dat iedereen om de twee jaar in een bepaalde regio naar een screeningsbus met röntgenapparatuur komt om gescreend te worden, moet je rekening houden met de capaciteit, het aantal laboranten dat je ter beschikking hebt, een bepaald aantal mensen uitnodigen. De foto gaat in een map en wordt doorgestuurd naar een radiodiagnost. Die beoordeelt ze, en ziet in sommige gevallen dat er mogelijk iets aan de hand is. Dan ga je een follow-up traject in, naar een ziekenhuis, een specialist enzovoorts. De applicatie bevolkingsonderzoek borstkanker (IBOB) ondersteunt dat hele traject, dus vanaf de ontvangst van de GBA-gegevens tot en met de inplanning, de uit-

nodiging, de brieven naar de vrouwen met een uitnodiging voor de screening, de afmeldingen en beoordelingen, en uiteindelijk ook het versturen van de brieven naar huisartsen, maar - veel belangrijker - ook de follow-up daarvan. De applicatie houdt dus ook in de gaten naar welk ziekenhuis iemand gaat, welke gegevens hieruit voortkomen, wat het uiteindelijke resultaat is. Uiteindelijk moet je ook weten of iemand terecht positief is, of dat het toch niets was. Dat is heel belangrijk, want als het terecht positief was, nodig je zo iemand normaal gesproken niet meer uit.'

## Koppelingen

De applicatie bestaat uit twee grote gedeelten. In het centrale deel wordt de planning, de uitnodiging en de verwerking geregeld. Er is ook een decentraal gedeelte.

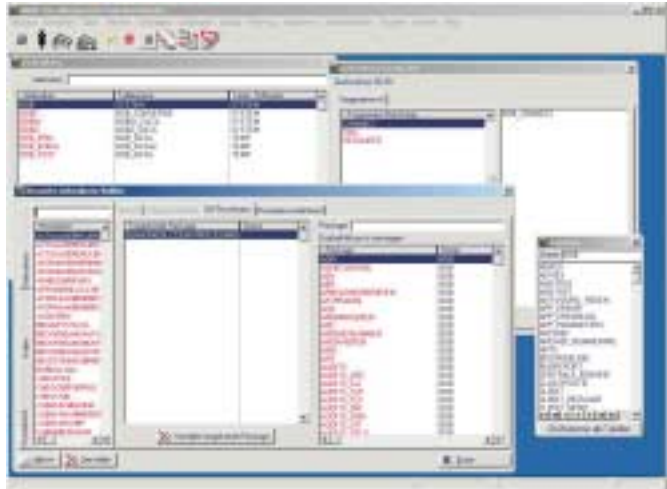
Pepping: 'Dat decentrale gedeelte zit op de units, dus in de screeningsbussen. Daar staat een pc met een applicatie die de diskette met de daglijst ontvangt. De lijst vermeldt alle mensen die op die dag gescreend moeten worden. Deze applicatie is gemaakt met Clarion en Oracle Lite. Voor diegene die Oracle Lite nog niet kennen: naar kijken! Dit is een fantastisch product voor stand-alone pc toepassingen. Er wordt vastgelegd hoeveel foto's er worden genomen en dat gaat vervolgens terug naar de centrale applicatie. Daarin zitten weer koppelingen naar andere administraties, voornamelijk de kankerregistratie, en naar het LETB, het Landelijk Evaluatieteam Borstkanker. Dit is een wetenschappelijke groep bij de Erasmus universiteit Rotterdam, die onderzoek doet naar de effectiviteit van het onderzoek. Deze mensen doen dat effectonderzoek in opdracht van de CVZ (College van zorgverzekeraars). Zij doen dus een totaalonderzoek naar het project, eigenlijk al sinds het begin. En er is nog een tweede partij: het LRCB, verbonden aan het Radboud ziekenhuis in Nijmegen. Die doen onderzoek naar kwaliteit van de screening. Ik vind dat in die zin het Nederlandse bevolkingsonderzoek heel goed is neergezet.'



Afbeelding 1. Het centrale deel van de applicatie regelt de uitnodiging, de planning en de verwerking

## Batchverwerking

Pepping: 'Het grootste deel van de applicatie is geschreven in PL/SQL. De front-end is een Clarion applicatie maar de hele processing, alles wat een proces is en iets verwerkt, zit in Oracle met PL/SQL. Dat is dus de hele functionaliteit van plannen, uitnodigen, verwerken en follow-up. Zelfs de afhandeling van brieven gebeurt in de database. Mailmerge files worden via



Afbeelding 2. Op het front-end zijn er buttons om de gebruikersautorisaties vast te leggen

PL/SQL aangemaakt en het front-end pakt die weer op en gaat met Word mergen. Eigenlijk is er maar één centraal systeem, want iedereen in heel Nederland gaat via ISDN en een VPN-netwerk naar die centrale omgeving toe. Die centrale omgeving is ingericht met Citrix en Oracle. Het printen van brieven gebeurt weer lokaal. Citrix spoolt dat naar locale pc's en LAN's.

Oracle kan natuurlijk wel brieven maken, maar het model kunnen we maken in Word en met mailmerge krijg je dan veel toegevoegde functionaliteit. Dat betekent ook dat de gehele security afhandeling in Oracle gebeurt. Op het front-end zijn er buttons om in te stellen wat welke gebruiker mag en wat niet - dat is geïntegreerd in Clarion (zie afbeelding 2). Hetzelfde geldt voor batchverwerking. Het inplannen voor een bepaalde regio gebeurt via batchverwerking. Dat is een gigantisch complex proces. Je moet met heel veel factoren rekening houden: zijn die mensen niet onlangs geweest? Hoe lang is het geleden dat ze zijn geweest? Hebben ze zich nooit afgemeld? Hoe zit het met de follow-up? Heel veel dingen worden in dat planningsproces bekeken. Dat is overigens allemaal PL/SQL en de gebruiker lanceert dat vanaf zijn front-end. Hij start het op en vervolgens wordt er met DbmsJob van Oracle ingescheduled. Dat kun je vertraagd doen of direct. Sommige zaken als het actualiseren van de doelgroepen en gebieden vinden 's nachts plaats - dat zijn zware processen -, maar het wordt via het front-end bestuurd. Dat is heel asynchroon gemaakt. Brieven

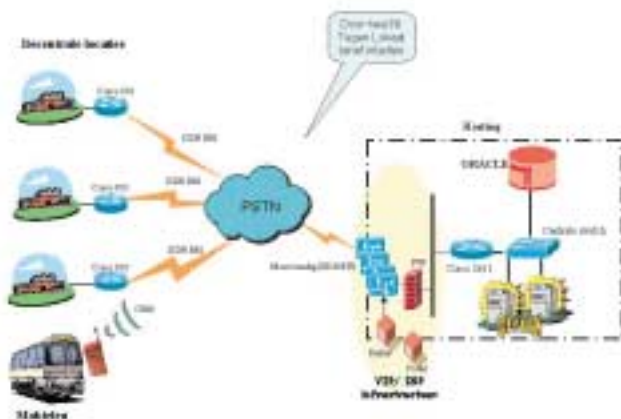


printen ook: dat kan op de achtergrond, het systeem meldt zich dan en zegt: ik ben klaar, start Word maar en dan Word kan mailmergen, dus helemaal asynchroon. Het mag synchroon, maar dat wordt zelden gedaan.'

## Stabiel

Een groot voordeel van een centrale applicatie is dat onderhoud en support veel eenvoudiger wordt.

Pepping: 'Doordat alles centraal is, kan de helpdesk heel gemakkelijk inloggen in de database als user x of y. Een nieuwe versie neerzetten is heel eenvoudig op deze manier. Oracle dba-achtige werkzaamheden doe je ook allemaal vanuit een centrale plek. In dit geval gebeurt dat bij ons in Maarssen. Je kunt op de Oracle server komen, je start dba studio en je kijkt of die taak hangt, of er iets mis is met een table space. Ik moet zeggen dat ik wel heel tevreden ben over Oracle, want het is heel stabiel. Het grote voordeel is natuurlijk: het systeem wordt echt als



Afbeelding 3. De configuratie van de remote access tot de bevolkingsonderzoekapplicatie

databaseserver gebruikt. Daar worden geen wijzigingen in aangebracht. De users worden aangemaakt in de Citrix servers. Dus die databaseserver is opgezet, en that's it, daar wordt verder niets meer aan veranderd.

Als je dat doet met Windows 2000, dan is het hartstikke stabiel. Maar op het moment dat er een systeemmanager aan gaat zitten, en die gaat service patches installeren en zegt, ach, weer eens een nieuwe driver installeren voor een of ander randapparaat, dan introduceer je steeds nieuwe bronnen van instabiliteit. Ik heb wel gemerkt dat bij het ASP-concept het aantal manieren waarop een systeem down kan gaan, groeit. Je moet goed leren om te identificeren waar het probleem ligt. Als iemand zegt: het werkt niet, waar ligt dat dan? Is dat dan lokaal, is het netwerk, is het de router - je hebt een hele trits van dingen. Is het de verbinding tussen de Citrix server en de Oracle server? Er zijn een heleboel componenten waar het procedureel mis kan gaan, maar tot mijn grote verbazing gaat er heel weinig mis.'

## Traceability

Pepping werkt graag met PL/SQL: 'Met PL/SQL kun je op een heel gestructureerde manier toch een object benadering doen omdat je packages kunt maken. Het is geen OO programmering, maar een package heeft wel zaken als encapsulatie. Je kunt er verschillende methoden in opnemen, je kunt ook aan information hiding doen, je kunt een package maken die 'uitnodigen' heet met verschillende functies en eigen data-elementen. We hebben ook een error package gemaakt, dat de hele error afhandeling doet. Errors worden gelogd in een tabel, en via werkingsverslagen weergegeven in HTML. Ook alles wat wordt gestart, wordt op die manier gelogd en bijgehouden. Dat betekent dat de traceability voor onze helpdesk heel hoog is. Zij kunnen gemakkelijk zien wat er gebeurd is, wat er gestart is en welke parameters er in het scherm opgegeven waren. Met Oracle kun je dat soort dingen allemaal op de server doen. Je kunt het heel centraal houden en het heel rigide opzetten. We hebben dan ook een heel goede performance, vind ik zelf. Het netwerkverkeer is minimaal. Je start iets met een paar parameters en je krijgt een resultaat terug. En dan kan het vervolgens worden bekeken.'

RADventure doet nu heel veel in PL/SQL, ooit zal dat, wanneer het aan Oracle ligt, Java moeten worden.

Pepping: 'Ja, ze zijn daar heel druk mee bezig om die overgang te maken, je hebt ook al tools en utilities van derden die PL/SQL in Java omzetten. Daar zijn volgens mij twee problemen mee: ten eerste is het aantal Java mensen dat dit soort dingen kan maken, nog heel gering op de markt. En het is veel complexer dan PL/SQL, omdat je dan ook ineens veel van objectoriëntatie moet weten. Je moet allerlei dingen includen om zaken aan het werk te krijgen, en je moet heel veel van de

Java API's weten. Bijvoorbeeld om een scherm aan te sturen zijn er tientallen API calls nodig. Terwijl je in PL/SQL zegt "declare cursor", en je begint te fetchen, dat is zo simpel. PL/SQL heeft een zo lange historie en heeft zich bewezen en de klanten zeggen altijd: "proven technology s.v.p.". Daar zit wel wat in. Maar ja, de Java-implementatie van Oracle, want de Java Virtual Machine zit in Oracle, is dat proven technology? Oracle zal natuurlijk roepen: "yes it is", maar ik weet het niet.'

## Platformonafhankelijkheid

Voorlopig gaat RADventure op de oude weg verder. Pepping: 'Rick van der Lans heeft laatst in zijn column in Computable geschreven over Java en C# met het .Net initiatief, en daar concludeerde hij tussen de regels door, dat .Net goedkoper en sneller is. Kijk, het grote voordeel van Java is dat het - denk ik - breder ondersteund is. IBM, Oracle en Sun staan erachter. Uiteindelijk, als je kijkt naar de coverage op platformen en operating systemen denk ik, dat je met Java beter zit, als je echt die portabiliteit wilt, maar je moet je natuurlijk wel afvragen als IT manager: "wil ik dat wel?" Het gebeurt eigenlijk niet zoveel dat mensen van operating system switchen, bijvoorbeeld van Unix, naar NT, heb ik het idee. Heb ik dat wel nodig? Wil ik wel zo leverancieronafhankelijk zijn? Kijk eens naar mensen die vroeger met Oracle begonnen, die zeggen, ik wil Oracle want dan ben ik platform en OS onafhan-

kelijk. Ik kan mijn applicatie zo van Unix naar NT porteren, of van Unix naar IBM AS/400 of VMS. Of op een Sun-box zetten, dus in die zin ben je redelijk ongebonden aan hardware en OS. Maar met handen en voeten gebonden aan Oracle, dat geef ik meteen toe. Je hebt natuurlijk Ansi SQL level 3, maar dat geldt natuurlijk niet voor mijn triggers, stored procedures en noem maar op. Het eieren eten is -op het moment dat je zegt: ik ga met PL/SQL aan de slag- dat je jezelf met handen en voeten aan een databasleverancier vastklemt. Sterker nog, als je zegt: ik ga met Java en Oracle aan de slag en je doet je implementatie van Java in je triggers en stored procedures, dan zit je ook met handen en voeten aan Oracle vast. Waarom wil je nu eigenlijk onafhankelijk zijn? Die vraag heb ik nooit echt kunnen beantwoorden. Wil je onafhankelijk zijn van je hardwareleverancier, van je operating system, van je databasleverancier, of onafhankelijk van degene die de taal produceert? Want uiteindelijk zijn er toch veel Java implementaties. Neem ik nu de Oracle Java developer kit of gebruik ik Forte van Sun? Tja...wie zal het zeggen. Ik vind, dat je als IT-manager ervoor moet zorgen dat je de service kan leveren, die de klant vraagt.'

**Tekst en fotografie: Dré de Man**

---