

CrossmarX won de RAD Race 2008. Daarvóór al de races van 2007 en 2006 en daarmee heeft het team bewezen met uiteenlopende opgaven en concurrenten te kunnen omgaan. Met hun 5GL zette het team de toon: het oprukken van framework-gebaseerde 5GL's ten koste van de 4GL's, 'kale' Java en C# zette zich in 2008 voort. Ruby bleek aan Rails niet genoeg te hebben.

RAD Race 2008: Een kortere route en minder CO₂

Winnaar CrossmarX:
'Alles bereikt wat we wilden bereiken'

Sommigen moeten nog bij donker van huis zijn gegaan. Bij aankomst bij het A4-hotel mengden ze zich tussen hotelgasten die op weg waren van of naar Schiphol. Misschien heeft een enkeling nog even gewenst in een van de overvliegende vliegtuigen te zitten. Een paar minuten later hield Werner Schoots een kort inleidend praatje waarop een uitleg volgde en vanaf dat moment was voor de deelnemers geen ontsnappen meer mogelijk. Tot de volgende dag 15.00 telde nog maar één ding: zo snel mogelijk een applicatie bouwen.

Voor de lezers van SRM is het een bekend gegeven: één keer per jaar zondert een groepje ontwikkelaars zich af om dat te doen wat ze hele jaar door al doen, maar dan in een wel zeer extreme vorm: programmeren. De RAD Race kent een lange traditie. Ooit geboren bij het inmiddels ter ziele gegane CM Corporate, daarna met wedstrijdleader Ivan Verborgh verhuisd naar Software Release Magazine. Na drie jaar continueerde Ivan die traditie in België terwijl SRM de eer in Nederland hooghield. Wedstrijdleaders volgden elkaar op, ook juryleden wisselden elkaar af, alleen voorzitter Rick van der Lans en Ron Tolido bleven de RAD Race door de jaren heen trouw.

Ook het concept van de RAD Race bleef in grote lijnen hetzelfde. In twee dagen moet een team van twee ontwikkelaars een goed werkende applicatie zien te bouwen. De applicatie lijkt zo veel mogelijk op administratieve applicaties zoals



Werner Schoots geeft het startschot voor de RAD Race 2008.

die in de praktijk ook gebouwd moeten worden. Die praktijk is in de loop van de jaren uiteraard veranderd. De eerste jaren lag het accent vooral op het betrouwbaar uitvoeren van transacties met wat beperkingregels binnen het toen voor de hand liggende kader van een fat client, ondersteund door een database. In latere jaren werd steeds meer de grenzen van de technologie opgezocht: de logica werd ingewikkelder, webapplicaties en webservices deden hun intrede en ook werden steeds hogere eisen aan het uiterlijk van de applicaties gesteld.

Voor de opgave van dit jaar gold dat zeker: van tevoren werd al bekend gemaakt dat bij de opgave GIS, en in mindere mate Bam en BI een rol zouden spelen, dat er gebruik gemaakt moest worden van webservices en dat er mashup-functionaliteit gevraagd zou worden.

Google of mijn webservice?

De afstanden tussen de verschillende verladens en vrachten werden niet in een tabel ter beschikking gesteld, maar moesten via webservices opgehaald worden of uitgerekend worden op basis van coördinaten. De Vervoerders en Verladens waren in een Google-map en Virtual Earth map aangegeven. In de tabel Vervoerders en verladens stonden de adresgegevens ook nog eens, met daarbij de coördinaten.



Google maps-weergave van vervoerders en verladens.

Uit de opdracht:

De onderlinge afstanden tussen de diverse vervoerders en verladens kunt u op verschillende manieren uitrekenen.

Google maps.

GRoute kent een methode `getDistance()`

<http://code.google.com/apis/maps/documentation/reference.html#GRoute>

Na registratie en het aanvragen van een API key via:

<http://code.google.com/apis/maps/signup.html>

kan daarmee gewerkt worden. De complete api is te vinden via:

<http://code.google.com/apis/maps/documentation/reference.html>

Met behulp van coördinaten kunnen afstanden berekend worden.

2. Webservices.nl biedt meerder webservices aan, waaronder een voor het opvragen van routes en afstanden. Speciaal voor de RAD Race was er een testaccount aangemaakt. Hun request ziet er zo uit:

```
POST /soap.php HTTP/1.0
Host: localhost:443
User-Agent: NuSOAP/0.7.2 (15907)
Content-Type: text/xml; charset=UTF-8
SOAPAction: "https://localhost/soap.php/routePlannerInformation"
Content-Length: 870
```

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://www.webservices.nl/soap/">
  <SOAP-ENV:Header>
    <HeaderLogin SOAP-ENV:mustUnderstand='1'>
      <username>redrace_User</username>
      <password>***</password></HeaderLogin></SOAP-ENV:Header>
    <SOAP-ENV:Body>
      <tns:routePlannerInformation xmlns:tns="http://www.webservices.nl/soap/">
        <postCodefrom xsi:type="xsd:string">2512DV</postCodefrom>
        <postCodeeto xsi:type="xsd:string">2402ZN</postCodeeto>
        <routetype xsi:type="xsd:string"></routetype></tns:routePlannerInformation></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

De response vervolgens zo:

```
HTTP/1.1 200 OK
Date: Tue, 10 Jun 2008 15:17:58 GMT
Server: Apache
Set-Cookie: ReactID=d4deb3795f935b9655612da076180966; expires=Wed, 10-Jun-2009 15:17:59 GMT; path=/; domain=ws1.webservices.nl
X-SOAP-Server: NuSOAP webservices.nl
Content-Length: 886
Connection: close
Content-Type: text/xml; charset=UTF-8
```

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://www.webservices.nl/soap/">
  <SOAP-ENV:Header>
    <nu7633:HeaderAuthenticate xmlns:nu7633="http://www.webservices.nl/soap/">
      <reactid xsi:type="xsd:string">d4deb3795f935b9655612da076180966</reactid></nu7633:HeaderAuthenticate></SOAP-ENV:Header>
    <SOAP-ENV:Body>
      <ns1:routePlannerInformationResponse xmlns:ns1="http://www.webservices.nl/soap/">
        <route xsi:type="tns:RouteInfo">
          <time xsi:type="xsd:int">2020</time>
          <distance xsi:type="xsd:int">36595</distance></route></ns1:routePlannerInformationResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

3. Verder kennen ook Microsoft en Oracle GIS-functionaliteit.

4. Tenslotte is het mogelijk om de afstanden handmatig op te vragen en in te voeren.

We gaan er in deze opdracht in ieder geval vanuit dat u op basis van de coördinaatposities of postcodes van Vervoerders en Verladens de routes weet te bepalen tussen beide en derhalve met de gegeven afstanden de totaalafstand van de routes weet te bepalen.



Anders dan in alle voorgaande jaren waarin de RAD Race onder auspiciën van SRM plaatsvindt, was de race niet gehuisvest door Caggemini. De vooraf geplande combinatie met een congres maakte de keuze voor een andere locatie noodzakelijk. Persoonlijk miste ik de sfeer en de hi tech uitstraling van Caggemini's ADC. Het is echter de vraag of de teams iets van het veranderde decor hebben gemerkt; die hadden vanaf het begin van de race vrijwel alleen oog voor hun schermen. Sommigen begonnen zelfs vrijwel onmiddellijk nadat de opgave werd uitgedeeld code te kloppen.

Grote lijnen

De grote lijnen van de opgave waren iedereen wel duidelijk: een bemiddelingsstelsel moest ervoor zorgen dat vrachtauto's die zonder lading onderweg waren, onderweg lading konden oppikken. Het meest ingewikkelde onderdeel ervan was de automatische planning: over een periode van een week moesten ca. 150 vrachten gecombineerd worden met ritten van vrachtauto's die anders een lege rit zouden maken - maar alleen als dat een zo groot mogelijke CO2-reductie zou opleveren. De noodzakelijke afstanden tussen de verschillende laad- en losplaatsen en beschikbare vrachtauto's stonden niet in een tabel maar moesten berekend worden, terwijl de gegevens via webservices opgehaald werden. Dat kon comfortabel via mijn-webservice.nl, of iets ingewikkelder via andere oplossingen als Google. Ook moest onder meer gerapporteerd worden hoeveel CO2-besparing

het systeem nu opleverde. Wie nog tijd en energie overhad kon een en ander via mashup's combineren met Google-kaartjes, zodat de applicatie uiterlijk net zo hip zou zijn als het onderwerp.

De gegevens die verstrekt werden bestonden uit lijsten van vervoerders en verladers met hun respectievelijke staanplaatsen, lijsten met vrachten die op een bepaalde tijd opgehaald moesten worden en lijsten met voor retourritten beschikbare vrachtauto's; en aanhangers met hun laadcapaciteit.

De opdracht bestond uit vijf onderdelen, waarvan er enkele inderdaad gemaakt konden worden zonder de opgave van a tot z door te lezen. Het eerste onderdeel was een basisadministratie, voor het beheren van verladers en vervoerders en het doen van financiële afrekeningen. Voor het merendeel eenvoudige schermen met wat CRUD-functionaliteit. Iedereen kan dit gemakkelijk bouwen, maar aan de andere kant waren er ook niet veel punten mee te verdienen: 5 van de in totaal 150 punten. Vreemd was dat slechts de helft van de teams voor dit eenvoudige onderdeel het maximale aantal punten haalde.

Scherm bouwen

Bij het tweede onderdeel – dat beloond werd met 10 punten – werd al meer gevegd van de inventiviteit van de ontwikkelaars. Er moest een scherm gebouwd worden met behulp waarvan het mogelijk gemaakt zou zijn om ritten op een zo gebruiksvriendelijk mogelijke manier te plannen. Zowel geplande ritten als openstaande vervoersorders moesten getoond worden, zodat deze gemakkelijk handmatig met elkaar te combineren zouden zijn. Ook moest het systeem bij het schenden van beperkingregels de gebruiker daarop wijzen. Veel teams waren daar nog in de middag van de eerste dag mee aan het worstelen. Bij dit onderdeel haalde geen enkel team het maximale

Overzicht modules en te behalen punten

1: Basisadministratie	(5 punten)
2: Handmatig plannen	(10 punten)
3: Automatisch plannen	(20 punten)
4: Inspelen op de actualiteit	(5 punten)
4.2	(10 punten)
4.3	(5 punten)
5: Managementinformatie	
5.1	(10 punten)
5.2	(10 punten)
5.3	(10 punten)
6: Presentatie	(20 punten)

aantal punten. De score beliep vaak 66 of 75 % procent van het maximum. De fouten hadden vrijwel steeds betrekking op niet helemaal geïmplementeerde beperkingregels.

Het derde onderdeel was behoorlijk gecompliceerd. De applicatie moest hier op een totaal van 150 ritten en 150 ladingen automatisch die combinaties zien te vinden waarbij de CO2-besparing het grootst was. Daarbij golden onder meer regels met betrekking tot het maximale gewicht van de lading en de tijd waarop die opgehaald en afgeleverd zou moeten worden. Veel deelnemers kwamen hier pas op de tweede dag aan toe, waardoor de tijdnood meteen heel nijpend werd, want om 13.00 zou gejureerd worden. Bij dit derde onderdeel waren de verschillen nog groter. Veel teams scoorden geen enkel punt, Servoy en Nedorce (team Roel) twaalf, 9 Knots en Novulo zestien en alleen Crossmarx de maximale 20 punten). Net als bij de opgave van vorig jaar en die van het jaar daarvoor bleek het geautomatiseerd maken van een keuze uit verschillende combinaties op basis van een algoritme een struikelblok. Niet alle IDE's (of ontwikkelaars) beschikten over de noodzakelijke SQL-vaardigheden en ook het schrijven van een algoritme blijkt ieder jaar weer moeilijker dan verwacht. Hierbij moet uiteraard ook aangetekend worden dat de tijdsdruk waaronder de deelnemers stonden een en ander ook compli-

ceerde. Waar in de praktijk een ontwikkelaar een algoritme van internet haalt, of een vraag bij een collega neerlegt, moesten de teams nu in korte tijd zelf met een oplossing komen. Aan de andere kant geeft dit soort programmaonderdelen al snel de beperkingen van een tool aan. Oplossingen waarbij de planning semi-automatisch verliep, waarbij de gebruiker dus meerder gegevens zelfstandig met elkaar moest combineren leverden geen punten op, ook al waren er teams die veel moeite deden uit te leggen dat automatisch soms ook kan betekenen dat de dingen niet vanzelf gaan.

Formule

De formule voor de CO2-uitstoot waarvan gebruik gemaakt werd, was in feite vrij simpel:

$(0,3 + (0,02 \times \text{laadgewicht in ton}) \times 2650 \text{ gram}$
 Als vermeld moesten alle afstanden in de opgave berekend worden op basis van gegevens die via een webservice opgevraagd moesten worden. Dit gold overigens ook al voor opgave twee. Meer hierover in het kader *Google of mijn webservice?* Bij de opdracht moest overigens ook het aantal kilometers getoond worden, dat gereden zou zijn wanneer er geen bemiddeling had plaatsgevonden. Voor de teams die dit soort gegevens gemakkelijk konden tonen, waren waarschijnlijk ook de managementrapportage (vijf) niet zo gecompliceerd.

Kijk op BPM-vendors

De BPM-matrix van Business Process Magazine

Bent u nog steeds op zoek naar een objectief marktoverzicht van BPM-tools?

De **BPM-matrix** van **Business Process Magazine** is een **onafhankelijk**, actueel overzicht van alle professionele software voor Business Process Management op de Nederlandse markt. Het is geen vergelijking, maar **objectief overzicht** van de functionele aspecten van de producten, zonder dat daar een waardeoordeel over wordt uitgesproken. Het gaat er immers om welk product het beste aansluit bij uw wensen en eisen. U kunt door de **BPM-matrix** browsen door te selecteren op leveranciers en op een kleine 100 functie-aspecten. Maar u kunt ook gebruikmaken van de **matchmaking module**, die aan de hand van door u ingegeven criteria voor u een shortlist samenstelt. Gewoon, om u te helpen een keuze te maken uit de vele uitstekende Business Process Management oplossingen die op de markt verkrijgbaar zijn. U bepaalt uiteindelijk zélf welke leverancier het beste bij u en uw bedrijf past.

U vindt de BPM matrix op www.businessprocess.nl



De BPM-matrix van Business Process Magazine staat onder redactie van senior consultants bij O&i te Utrecht.

**business
process
magazine**

Voor de teams daaraan toe kwamen, werkten de meeste echter aan opgave vier: inspelen op de actualiteit. Net als bij opgave drie waren daar (in totaal) twintig punten mee te behalen. In de praktijk bleek ook dat teams die zowel drie als vier gecompleteerd hadden bij de beste vier behoorden. Uitzondering was het tweede Ruby-team van Nedforce (team Roel). Dit team scoorde hier net zo goed als de winnaars, maar had veel andere onderdelen niet af, wat heel jammer was.

Opricht vier draaide om het idee dat vanaf nu op willekeurige momenten vervoersopdrachten aangeboden konden worden. Deze opdracht viel uiteen in drie onderdelen, de extra vervoersopdrachten en –aanbiedingen die verwerkt moesten worden (10 punten), het bouwen van schermen om extra vervoersaanbiedingen via e-mail of sms te verwerken (5 punten) en het generen van een XML-bericht dat de verlader op de hoogte brengt van de status van de extra binnengekomen vervoersopdracht. Om dit goed te controleren werd pas op dag twee een extra serie vervoersopdrachten aangeleverd.

Changes

Onderdeel vijf van de opdracht betrof de managementinformatie. Er werd zowel om rapportage in tekstuele als in grafische vorm gevraagd, waarbij bovendien nog de eis gesteld werd dat de grafieken de rapportage zo duidelijk mogelijk zouden maken. CrossMarx en Hot Item scoorden hier het maximale aantal punten (30), Servoy 25 en Novulo 15. De andere teams waren aan dit onderdeel nauwelijks toegekomen, scoorden in ieder geval nauwelijks punten.

Het spannendste onderdeel van de race (ook het leukst om te verzinnen) betreft de *changes*. Het idee hierbij is dat met het veranderen van de applicatie duidelijk zou moeten worden hoe goed deze te onderhouden is. Er zijn altijd veel punten mee te verdienen. Dit jaar was er maar één change, althans is er maar een change echt uitgedeeld. De wijziging bestond daarin, dat de vervoerders 24 uur extra kregen voor het transport. In feite een vrij kleine verandering van de logica. Voor de jurering had deze aanpassing bovendien het voordeel, dat heel gemakkelijk te controleren zou zijn wie deze doorgevoerd zou hebben. 9 Knots en CrossMarx scoorden hier de maximale 20 punten, Novulo en HOT Item 15. Verder had niemand deze *change* doorgevoerd.

Jurering

Voor de meeste teams kwam de jurering te vroeg. De juryleden controleerden ieder een onderdeel van de opdracht met behulp van juryscripts. Er werd vooral gekeken of de gebouwde applicaties de in de opdracht gevraagde functionaliteit bezaten. Zoals uit de grafiek blijkt (zie grafiek eind-

resultaat) verschilde dat nogal. Functionaliteit ontbrak doordat de applicaties nog niet af waren, onderdeel vijf ontbrak vaak en aan de changes waren alleen de beste vier toegekomen.

De vier teams die het hoogste aantal punten bij de jurering met juryscripts behaald hadden, kwamen ook in aanmerking voor de zogenaamde *Idols*-jurering: hierbij werd de teams ten overstaan van publiek gevraagd hun applicaties te tonen terwijl ze streng ondervraagd werden. Daarbij stond één vraag vast: de deelnemers moesten een extra vervoersaanbod invoeren en een extra vervoerder, waarop de - nu veranderde - uitkomst van het automatische plannen gecontroleerd werd. Deze test stelde Hot Item al snel voor vrijwel onoverkomelijke problemen. Ook 9Knots had er moeite mee.

Winnaar CrossMarx bleek zonder enig probleem de wijziging door te kunnen voeren net als nummer twee, Novulo. Aangezien deze twee een iets ander algoritme gebruikt hadden verschilden de uitkomsten. Na berekeningen via de PDA-applicatie van Wouter Goedvriend en discussies over de aard van het algoritme met Cor Baars kwam de jury tot de conclusie dat de uitkomst van Novulo minstens zo goed was als die van CrossMarx. Ook verder bleek Novulo heel veel punten binnen te halen bij de *Idols* jurering, vooral door de duidelijkheid en uitvoerigheid van de managementinformatie. Uiteindelijk kreeg Novulo er 43 (van de 45) punten bij, Crossmarx 26, 9 Knots 15 en Hot Item 6. Daardoor haalde Crossmarx weliswaar de overwinning, maar met een iets minder grote voorsprong als voorgaande jaren. Novulo zou het volgend jaar wel eens gemakkelijk kunnen krijgen, vooral omdat Crossmarx dan niet meer meedoet. Na drie overwinningen achter elkaar heeft het team alles bereikt wat ze wilden bereiken.

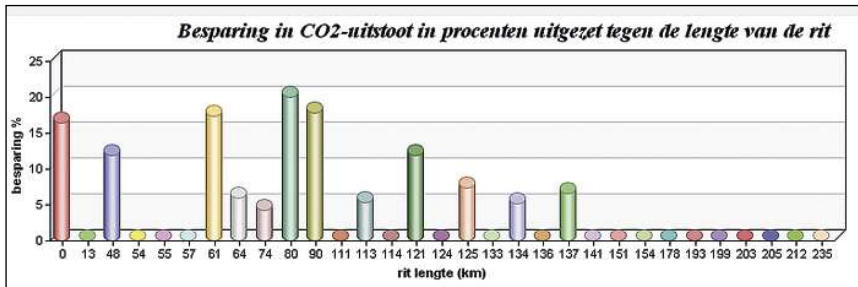
Man, paard en jury

De jury bestond uit IT-goeroes **Rick van der Lans**, Directeur van R/20 Consultancy (juryvoorzitter) en **Ron Tolido**, CTO Continental Europe & Asia Pacific bij Capgemini, **Cor Baars** (DNV-CIBIT), **Chris Widdows** (Ordina), **Daan Calmeijer** (DNV), **Graham Bolton** (IFSQ), **Wouter Goedvriend** (Seven Stars) en **Dré de Man** (Array Publications). Extra inzicht in de aard van de code gaven dit jaar codereviews van IFSQ. Het concept van de opgave werd gemaakt door Cor Baars, terwijl Dré de Man de opgave verder afgemaakt heeft en Wouter Goedvriend de juryscripts opleverde.

De uitslag (zie grafiek) is berekend ná de openbare ('*Idols*-') jurering. Daarvóór, uitsluitend op basis van de functionaliteit van de applicaties, was de voorsprong van CrossmarX nog groter.

Tools en teams

1 CrossMarX haalde met een slim framework van Java-classes de eerste plaats. Hun tool zou je een 5GL kunnen noemen, of een open 4GL: een ontwikkelomgeving die het voordeel van het gebruiksgemak van een 4GL combineert met dat van de openheid van talen als Java. Het tool werkt met een server, de CrossmarX Application Engine, in feite een verzameling Java-classes (pojo's: plain old Java objects) die in een TomCat applicatieserver draaien. Die classes zitten in een repository en kunnen aangepast worden en desgewenst uitgebreid met plug ins, ook Java-classes. Uiteindelijk ontstaat er ook HTML, SQL, en Javascript. Delen van de CrossmarX Application Engine zijn gepatenteerd. CrossmarX claimt dat ook niet-ontwikkelaars na een paar uur met hun tool kunnen werken.

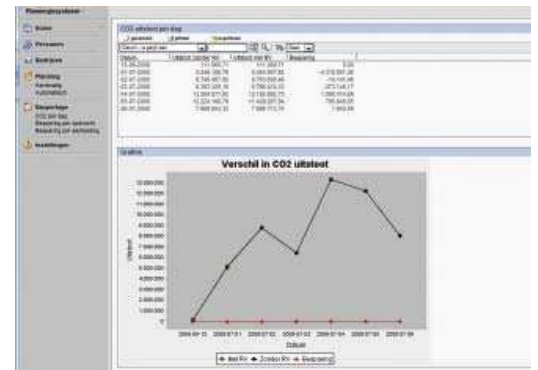


CrossmarX Managementrapportage.

2 Tweede werd Novulo met hun 'Visual Ajax Builder' die overigens behalve behalve Javascript en XML vooral C# genereert. In plaats van C# kan ook voor PHP gekozen worden. Bij navraag blijkt Novulo de term Ajax Builder niet meer zo geslaagd te vinden, liever noemen ze het tool een software development system dat het ontwikkelen, deployen en onderhouden versnelt. Het werkt een beetje als een 4GL, in die zin dat je begint vanuit de user interface. Van daaruit



Rick van der Lans (l) bij de jurying van het Novulo-team.



Novulo verschil CO₂-uitstoot planning.

ontstaat functionaliteit en wordt eveneens een datamodel gecreëerd. Het tool ondersteunt open standaarden waaronder de complete DHTML stack. Novulo is Microsoft partner.

Zoals het een Visual Ajax Builder betaamt leverde die een applicatie op die ook zeer goed oogde, zodat het team tijdens de openbare Idols-jurering extra punten scoorde.

3 9 Knots en HOT Item deelden de derde plaats. 9 Knots gebruikte het eigen product Casemaster (een applicatieontwikkelframework gebaseerd op het principe van zelfbeschrijvende objecten), in feite een variant op een interpretator. Casemaster produceert bytecode, tijdens de wedstrijd VB 6 bytecode, maar het kan ook Java bytecode ophoesten.

3 HOT Item had in ieder geval politiek gezien de interessantste oplossing: het werkte met een set van open source frameworks voor Microsoft C# (Castleproject, vooral Monorail MVC framework en Active record) in combinatie Microsoft Visual Studio. Een van de teamleden, Roelof Blom, is overigens erg actief binnen <http://www.castleproject.org>.

Servoy scoorde bij de jurying op basis van juryscripts net te weinig punten om aan de openbare jurying mee te doen. Het team scoorde nergens slecht, alleen hier en daar soms net iets minder dan de allerbesten. Wat Servoy echt de das om gedaan heeft, is dat het niet aan de change toegekomen is, want dat zou twintig punten extra opgeleverd zou hebben. De rapportage van de managementinformatie had Servoy nu juist weer bijzonder goed gedaan. Het team had erop gegokt bij de beste vier te eindigen. Zou dat gelukt zijn, dan had de strategie goed gewerkt. Jammer, maar

strategie is een belangrijk onderdeel van de race, trouwens ook bij het binnenhalen van projecten.

Servoy is een interpreter op basis van Java. Bijzonderheid van dit jaar was dat Servoy een vrouwelijke ontwikkelaar inzette, de jongere zus van degene die de jaren daarvóór de lat hoog moest houden. De eerste vrouwelijke ontwikkelaar tijdens de RAD Race was overduidelijk geen Excustruus. Bij een volgende editie zou ze haar broer wel eens kunnen overtroeven, nu evenaarde ze hem in ieder geval. Servoy doet nu voor de derde keer mee, heeft nooit gewonnen maar weet iedere keer een behoorlijke score neer te zetten, wat op zich wel een prestatie is.

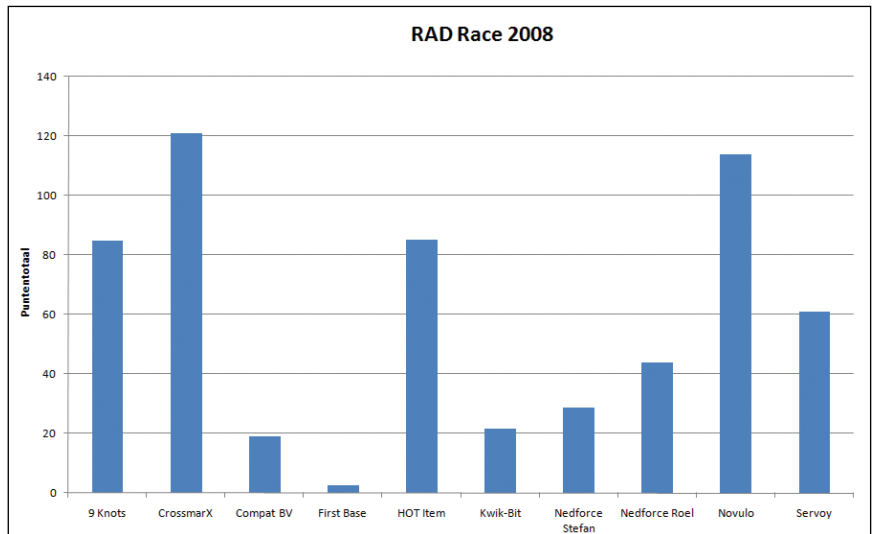
Opmerkelijk was dat deze keer Ruby on Rails slechts middelmatig presteerde. De beide zeer jonge teams van Nedforce wisten vooral op de eerst dag niet voldoende functionaliteit te bouwen en haalden die achterstand op de tweede dag niet echt in. Voor het belangrijkste deel kwam dat doordat ze anders dan de RoR-teams van voorgaande jaren, geen gebruik maakten van frameworks voor het bouwen van CRUD-functionaliteit (CRUD = create, update delete, basisbewerkingen met data). Ook waren de teams wat minder ervaren dan de Rubianen van voorgaande jaren.

Kwik Bit werkte met WinDev, een Franse ontwikkelomgeving met een eigen taal (WLanguage, een soort mix van Basic en DBase) die ook Ajax en web-services produceert en een rapport-generator heeft. Theoretisch een geschikte kandidaat, die echter om welke reden dan ook niet genoeg functionaliteit produceerde, met name op de tweede dag.

Compat werkte met een oplossing die een paar jaar geleden heel hip was, maar die ook nu nog goede resultaten op zou moeten kunnen leveren:



De RoR-teams maakten geen gebruik van frameworks voor het bouwen van CRUD-functionaliteit. Zeer waarschijnlijk zijn ze daardoor niet bij de eerste vier beland.



Voor het belangrijkste deel kwam dat doordat ze anders dan de RoR-teams van voorgaande jaren, geen gebruik maakten van frameworks voor het bouwen van CRUD-functionaliteit.

UML om prototypes in .Net (2.0) op te leveren. Het produceerde fraaie modellen maar te weinig functionaliteit om echt goed te scoren, misschien ook wel doordat het team nogal schrok van de harde werkelijkheid van de RAD Race.

Het team met de beste RAD Race spirit was ongetwijfeld First Base. Onvoorbereid maar wel met twee nieuwe hard disks en één nieuw besturingssysteem (Ubuntu) trad dit jaar ten strijde. Met name het niet-functioneren van de nieuwste versie van hun ontwikkelomgeving op Ubuntu leverde zoveel achterstand op, dat dit team aan het einde van de eerste dag inzag dat verder strijden geen zin had.

Alle tools die meedoen met de RAD Race proberen dat nadeel te ondervangen door een 4GL-achtige oppervlakte te combineren met een van de moderne talen als Java of C#. Ze gebruiken deels bestaande code, en/of produceren code of bytecode. Een zekere zin een uitzondering vormt Ruby on Rails, net als Java en C# een OO-taal, maar met een eenvoudiger syntax, een eigen framework (Rails) en regels die neerkomen op het werken met standaardoplossingen en het aangeven van uitzonderingen daarop.

De ontwikkelingen binnen de RAD Race zijn echter ook terug te vinden binnen de Java- of Microsoft .Net wereld. Frameworks en ander hulpjes zorgen ervoor dat het programmeerwerk langzaam maar zeker sneller gaat. In die zin is de RAD Race vergelijkbaar met de Formule 1, waar ook technologie wordt gebruikt die pas later zijn toegang tot seriemodellen vindt.