

# Het opzetten van een Open Source Ontwikkelstraat

met behulp van Java tools...

## Agenda

- Achtergrond
- Opdracht
- Uitgangspunten
- Tools
- Problemen
- Lessons learned

## Achtergrond

- Mijn achtergrond
  - Software Development
  - Interesse in Open Source en Open Source Java
  - In Eindhoven de enige... (toen!)
  - Gesprek met klant

## Opdracht

- Aantonen dat een Open Source Ontwikkelstraat mogelijk is, en dat er snel gebruikersapplicaties op gebouwd kunnen worden.
- Applicatie bestond al, maar was gebouwd in MS Access, kon maar op een locatie gebruikt worden, en de versie van Access werd niet meer ondersteund...

## Uitgangspunten

- Opdrachtgever:
  - Moet draaien op Windows NT
  - Database SQL Server
  - Geen investeringen in hardware
  - Zo kort mogelijke doorlooptijd
- Team:
  - Java: een senior, vier juniors
  - Open Source: weinig kennis

## Uitgangspunten -2-

- Iedereen begint tegelijkertijd...
- Dus: eerst developeromgeving inrichten, mensen “bezig houden” (inlezen ontwerp, Struts, voorbeeld schermen bouwen)
- Dus: alles moest snel, weinig tijd om na te denken – iets wat zich later zou wreken

- Developer-omgeving:
  - Eclipse
    - Sysdeo plugin (Tomcat)
    - Easy Struts plugin (Struts)
    - Checkstyle plugin
    - XMLBuddy plugin
  - Struts
  - Hibernate
  - Log4J
  - JUnit
  - Tomcat
  - Poseidon CE

- Developer-omgeving:
  - Een aantal tools, verschillend per developer (“leken wel handig”)
  - Dit leidde tot grote verscheidenheid, waarvan de verschillen in Eclipse instellingen tot de grootste problemen leidde
  - Zelfs eigen lay-out directory-structuur
  - Geen beeld van werking CVS... laat inchecken, toch werken op eigen kopiën

- Integratie-omgeving
  - CVSNT
  - Maven
  - CruiseControl en Ant
  - Scarab en MySQL
  - JMeter
  - Tomcat
  - RealVNC
  - Custom-build bat-files (mijn “haarlemmerolie”)

- Testomgevingen
  - Draaiden op een machine (desktop ☹)
  - Een Tomcat installatie
  - Drie verschillende instances (dev, test, acc)
  - Wel een aparte SQL Server testmachine
    - Maar 5 concurrent connecties mogelijk
- Productieomgeving
  - Tomcat
  - Aparte SQL Server productiemachine.

- Moeilijk plannen
  - Als het mee zit, 3 uur, als het tegenzit misschien wel 3 weken... - installatie van een tool is niet het probleem, maar de configuratie om het naadloos met de rest te laten samenwerken...  
Doordat het soms tegenzat, te weinig tijd voor fine-tuning scripts.

- Voor de eerste keer gebruik in “productie-omgeving”
  - Zo begonnen we bijvoorbeeld met Anthill voor CI – maar dat kregen we niet aan het werk, dus gingen we maar over op CruiseControl (wat trouwens een prima tool is!)
- Geen CVS toegang

## Problemen - 3 -

- CVS / CVSNT
  - Bij het gebruik van CVS dient rekening gehouden te worden met het feit dat veel mensen gewend zijn om programma's, classes en dergelijke uit te checken, en die dan exclusief in gebruik te hebben. CVS gebruikt een optimistisch team model, en stelt iedereen in staat om zaken uit (en in) te checken, mits geautoriseerd. Hierdoor loopt men het gevaar dat twee mensen tegelijkertijd dezelfde file bewerken (en inchecken).
  - Mits er goed gecommuniceerd wordt levert dit zelden problemen op, Zorg er wel voor dat binnen een team slechts een iemand de verantwoordelijkheid draagt over CVS, en dat deze persoon de nodige bevoegdheden heeft.
  - Wanneer CVSNT gebruikt wordt in combinatie met Eclipse, moet de "Repository prefix" uitgeschakeld worden.
  - Sobig virus verspreid via mailing-list...

- Tomcat
  - Gebruikte versie vereiste een patch
  - Op testomgeving (1 PC, meerdere instances - voor dev. test, test en acceptatietest) niet installeren als service
  - Vaak werkte hot deployment niet (goed). Batch file geschreven, dat Tomcat stopte, alle delen van onze applicatie van een instance verwijderde, de war-file kopiëerde en tomcat opnieuw opstartte

- Scarab
  - Scarab was nog in beta, maar configureerbaar en stabiel.
  - Er waren twee goede alternatieven: Bugzilla en Jira.
  - Achteraf voor de grootte van het project eigenlijk te “bewerkelijk”, en te zwaar!

- Omdat veel developers hun eigen tools (en daardoor hun eigen configuratie gebruikten), moesten in de ANT en MAVEN scripts veel trucs uitgehaald worden om op de testserver alles synchroon te krijgen. Dit kostte enorm veel tijd!

## Lessons learned

- Zorg voor een eenduidige ontwikkelomgeving (directory-structuur etc.)
- Zorg voor de bevoegdheid om het gebruik van bepaalde tools af te dwingen of te verbieden
- Neem de tijd om mensen CVS te laten begrijpen

## Lessons learned - 2 -

- Automatiseer alles wat je kunt automatiseren (batch-scripts)
- Zorg eerst dat iets draait. Finetunen kan altijd nog
- Documenteer (aannames, keuzes, wijzigingen, scripts)
- Leer van je fouten!

- Inmiddels draait de Ontwikkelstraat een aantal maanden, en de applicatie draait – probleemloos – in productie!
- Wij zijn nu bezig deze ontwikkelstraat uit te breiden...