



Basis Project Tools voor Java Development Teams

Cemal Karacoban

Leon de Kaper

Maarten Baas



- Cemal Karacoban:
 - Java/JEE Consultant
 - Atos Origin Emerging Technologies
- Maarten Baas:
 - Java/JEE Consultant
 - Focus op productivity en Open Source
- Leon de Kaper:
 - Java/JEE Consultant
 - Innovatie projecten binnen Atos Origin Emerging Technologies

.nl. jug



Agenda

- Inleiding
- Concepten:
 - Version control system
 - Issue tracking
 - Build scripting
 - Testen
 - Quality Assurance
 - Continuous integration
- Projecttools:
 - Valkuilen
 - Orchestratie/Samenwerking
 - Project Omgeving in de praktijk
- Vragen

.nl. jug Inleiding

- Veel voorkomende problemen:
 - Veel handmatige acties (build, test & deploy)
 - Onduidelijke werkzaamheden (wie, wat & waarom)
 - Voortgangsbewaking
 - Kennisoverdracht
 - 'Mysterieuze bugs'



Inleiding

- Interface component van A en B
- Toetsen van componenten en requirements
- Onder druk vervallen controles



.nl. jug



Inleiding

- Fouten voorkomen
- Op tijd testen
- Kleine stappen testen

Test
early,
test
often

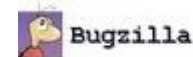




Inleiding



Project Tooling



.nl. jug Concepten

- Version control system
- Issue tracking
- Build scripting
- Testen
- Quality Assurance
- Continuous integration



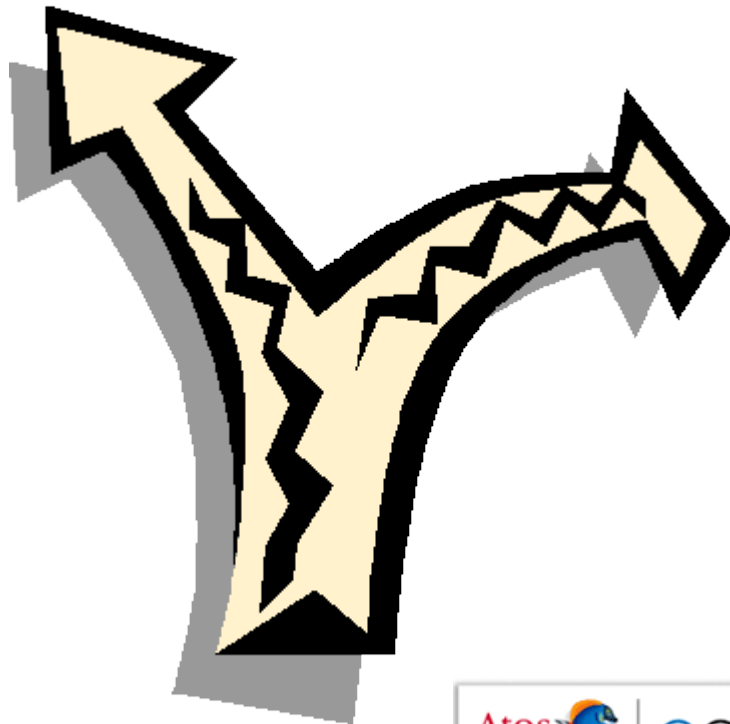
Version control system





Version control system

- Alle code centraal bewaard
- Versioneren:
 - Code
 - Releases
- Parallel:
 - Code
 - Releases





Version control system

- CVS:
 - Nog steeds meest gebruikte opensource VCS
 - Per bestand wijzigingen bijhouden:
 - Elk bestand een eigen versie nummer
 - Moeilijk wijzigingen achter te halen
 - Trager
 - <http://www.nongnu.org/cvs/>
- Subversion:
 - Wint snel aan populariteit, de keus bij nieuwe projecten
 - Houdt per wijziging de bestanden bij:
 - Makkelijk om wijzigingen achterhalen
 - Ook directories version-controlled
 - Sneller
 - <http://subversion.tigris.org>
 - <http://svnbook.red-bean.com/>



Issue tracking

Een centrale plaats voor het vastleggen, opsporen en bijhouden van:

- Taken
- Fouten
- Discussies
- Processen





Issue tracking

- Bugzilla:
 - Bij veel projecten gebruikt (Mozilla, Apache, Eclipse)
 - Moeilijk installeren en onderhouden
 - Niet gebruikers vriendelijke interface
 - <http://www.bugzilla.org/>
- Trac:
 - Web-based project management tools
 - Wiki omgeving
 - Online repository bekijken, bv. bekijken van sources in subversion repositorie
 - <http://trac.edgewall.org/>



Build Scripting (1)

- Bij het opzetten van een project moet altijd een aantal onderdelen gespecificeerd worden, namelijk:
 - Waar staan mijn bronbestanden?
 - Waar moet de uitvoer (class-files) naartoe?
 - Welke andere taken moeten er (eventueel) nog gedaan worden, zoals bijvoorbeeld het maken van een jar-file of het uitvoeren van unit tests?



Build Scripting (2)

- Ant:
 - Geschreven in Java
 - Task based
 - ANT bevat een uitgebreide set van ingebouwde taken zoals:
 - Java-compilatie en CL execution
 - De manipulatie van files en directories
 - Het maken van jar files
 - Het oproepen van CVS-commando's
 - <http://ant.apache.org/>



Build Scripting (3)

- Maven:
 - Gebruikt declaratieve manier i.p.v. task-based benadering
 - Heeft duidelijke standaarden
 - Beschrijft project structuur en inhoud
 - Veel meer dan build scripting
 - Dependency Management
 - Genereren documentatie en projectsite
 - <http://maven.apache.org/>



Testen

- **Waarom:**
 - Fouten voorkomen!
 - Kosten en tijd besparen
 - Opleveren betrouwbare programmacode
- **Aantal testtechnieken:**
 - Unit testen
 - Load & Stress testen:
 - Load: systeem op representatief niveau belasten
 - Stress: systeem boven nominale belasten
 - Regressie testen:
 - Testen van het ongewijzigde deel





Testen (1)

- Unit test:
 - Een methode om stukjes broncode (units) afzonderlijk te testen op een correcte werking
 - Een belangrijk middel om te bepalen of de code dat doet wat ervan verwacht wordt
- In de praktijk:
 - Een programma die de publieke interfaces van de classes test
 - Methoden doen wat je ervan verwacht
- Voordelen van Unit Testen o.a.:
 - Bij het aanpassen of refactoren van software
 - Bij het integreren van softwaremodules



Testen (2)

- Junit:
 - Unit test framework, <http://www.junit.org/>
- Selenium:
 - User interfaces testen, <http://selenium.openqa.org/>
- SoapUI:
 - Web services testen, <http://www.soapui.org/>
- JMeter:
 - Load testen, <http://jakarta.apache.org/jmeter>

<http://www.opensourcetesting.org/>



Quality Assurance (1)

- Doel:
 - Schrijven van goede, betrouwbare, flexibele, onderhoudbare code
- Problemen:
 - Verschillende meningen, manieren en gewoontes
 - Verschillende expertise
 - Etc.



Quality Assurance (2)

- Checkstyle (Code Standards):
 - Code standaarden afdwingen
 - IDE integratie
 - Build script integratie
 - Default Sun Code Conventions
 - Class design problems
 - Duplicate code



Quality Assurance (3)

- PMD (Static Code Analyse tool):
 - Ontdekt potentiële fouten en 'unsafe' code
 - Configureerbare 'rules'
 - Ontdekken kopie en paste
 - IDE integratie
 - Build script integratie



Quality Assurance (4)

- FindBugs (Potentiële Bugs):
 - Bytecode i.p.v. source code
 - Focus op performance en potentiële fouten
 - Gebaseerd op bekende fouten
 - Voorbeeld:
 - Adres adres = klant.getAdres()
 - if(adres !=null || adres.hasPostCode())
 -
- Wat is hier aan de hand???



Continuous integration (1)

- Praktijk problemen:
 - Hoe langer je wacht met integreren hoe moeilijker het wordt:
 - Denk aan conflicten en niet werkende functies
 - Unit test lost niet alle integratie problemen op
 - Problemen worden erger als deadline krappert
 - HELL 1:
 - Je haalt de code uit repository maar het werkt niet meer
 - HELL 2:
 - Je weet (of denkt) dat het niet aan jouw code ligt
 - Iedereen zegt “Het ligt niet aan mijn CODE” ☹️
 - WELKOM in Integratie HELL



Continuous integration (2)

- CI is een software ontwikkeling praktijk waarbij:
 - Elke ontwikkelaar minmaal een keer per dag zijn werk met de bestaande software integreert
 - Een programma of tool controleert automatisch op integratie fouten





Continuous integration (3)

- Continuüm:
 - Makkelijk te installeren en gebruiken
 - <http://maven.apache.org/continuum>
- CruiseControl:
 - Oudste CI tool, flexibel in gebruik
 - <http://cruisecontrol.sourceforge.net/>
- Hudson:
 - Een nieuwkomer, goede GUI
 - <http://hudson.dev.java.net/>



Project Tools

- Toepassen
- Samenwerking





Project Tools - Valkuilen

- Valkuilen bij implementatie:
 - Project tooling is een **MIDDEL** maar geen **DOEL!!**
 - Big bang
 - Slechte koppeling
 - Onjuist gebruik





Concepten Orchestreren

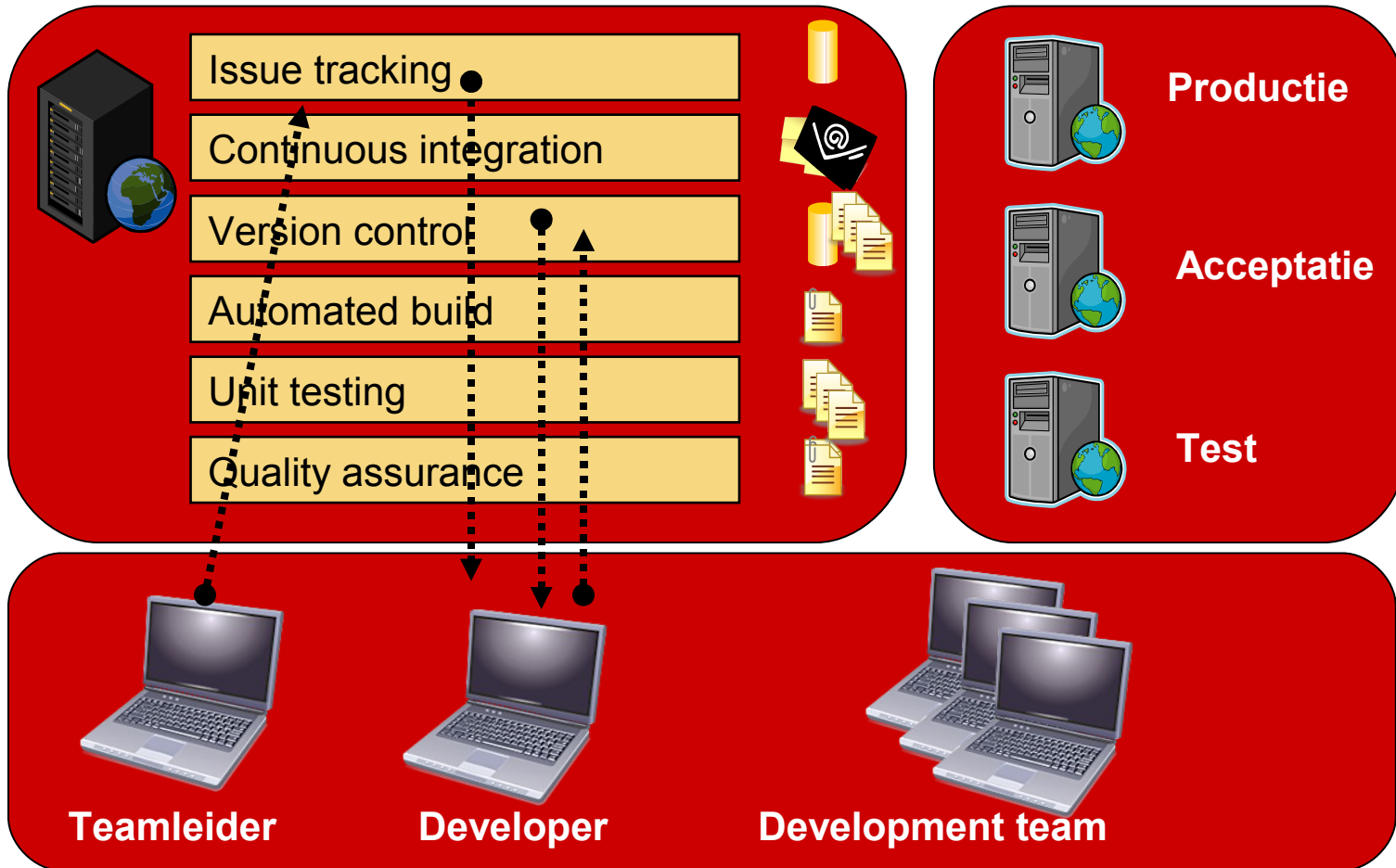
- Neem tijd voor het inrichten
- Maak iemand verantwoordelijk
- Documenteer
- Controleer



Zoveel mogelijk automatiseren!



Voorbeeld Project Omgeving (DEMO)



.nl.
jug



Conclusie

- Project tooling helpen bij projectuitvoering
- Afspraken en richtlijnen vastleggen
- Voor meer informatie:
 - Application Development & integration Blog:
 - <http://adi.atosoriginblog.nl/>
 - Buildix (opensource PMT):
 - <http://buildix.thoughtworks.com>

.nl.
jug



Vragen

