How to: proftaak in ontwikkelstraat hangen

**Inleiding**

Een ontwikkelstraat is een serie van tools die je helpt bij het ontwikkelen van je applicatie. Je kunt hierbij denken aan:

* een version control systeem (e.g. SVN of GIT) dat je helpt bij het bijhouden van wijzigingen op de broncode
* een tool dat de aanwezige tests uitvoert en feedback geeft over code coverage
* een tool dat analyse doet van je broncode en feedback geeft over de kwaliteit van de code (e.g. SonarCube)
* een tool dat de code uitrolt op productie servers

De kracht van de ontwikkelstraat is dat die volledig geautomatiseerd wordt uitgevoerd in de achtergrond. Dat betekent dat de ontwikkelaar heel vaak en heel snel feedback krijgt op zijn werk, waardoor eventuele problemen heel snel ontdekt worden en heel snel opgelost kunnen worden. In het Engels wordt een ontwikkelstraat “continuous integration” of “continuous delivery” genoemd.

Voor onze ontwikkelstraat maken we gebruik van het tool Jenkins. Jenkins draait op een Avans server met ip adres 145.48.6.147. Als je je broncode aanpast en de aanpassing commit in SVN of GIT, dan merkt Jenkins de commit op, kopieert de gewijzigde code naar de Avans server, compileert de code, voert de aanwezige tests uit en roept alle tools aan die in de ontwikkelstraat zijn gehangen. In ons geval, hangt er maar één tool in de ontwikkelstraat, namelijk SonarCube.

**Noodzakelijke aanpassingen aan de proftaak broncode**

Om het bovenstaande mogelijk te maken, zijn er een aantal aanpassingen nodig aan je Netbeans of Eclipse project:

* het project moet voldoen aan Maven projectstructuur (zie onder voor meer details)
* de broncode moet in een version control systeem staan dat bereikbaar is voor Jenkins op de Avans server. Dit kan je eigen installatie van SVN of GIT zijn, maar makkelijker is het om een gratis cloud systeem te nemen, zoals [assembla](http://www.assembla.com/) voor SVN, of <https://bitbucket.org/> of <https://github.com/> voor GIT. **Let op**: zet alleen de src/, pom.xml en nbactions.xml onder version control.

**Je project in de Maven projectstructuur zetten**

Je kunt in Netbeans of Eclipse direct een project volgens de Maven projectstructuur aanmaken. Heb je dat niet gedaan, dan staat hieronder hoe je een bestaand project kunt aanpassen.

1. Zet je bron- en testcode in de onderstaande directorystructuur. De bestanden nbactions.xml en pom.xml staan los op BB:

<projectnaam>/src/main/java/<packages met broncode>  
<projectnaam>/src/test/java/<packages met testcode>  
<projectnaam>/nbactions.xml  
<projectnaam>/pom.xml

Doe dit overigens niet met de “refactor” functie van Netbeans of Eclipse, maar in de Windows verkenner, want de packages moeten niet mee veranderd worden. Voorbeeld: het bestand /carapp/src/main/java/domain/Car.java gaat over de

klasse Car in het package domain.

1. De volgende aanpassingen zijn nodig om de proftaak instanties van de verschillende studenten of profgroepen naast elkaar te kunnen laten draaien:

* Pas in pom.xml <artifactId> aan naar bijvoorbeeld "HartigeHapErco". Hierdoor beschouwt Maven het als een ander artifact (en daardoor ook Jenkins, omdat Jenkins Maven gebruikt).
* Pas in het pom.xml ook <name> aan naar bijvoorbeeld " HartigeHapErco". <name> wordt als projectnaam gebruikt door Sonar. Zo kun je gemakkelijk zien welk Sonar project bij welke applicatie hoort.
* Pas in nbactions.xml bij de tag <exec.args> de Main klasse aan, naar de Main klasse waarmee jouw applicatie gestart wordt.
* Creëer, lokaal op de Avans server, een MySQL database, bijvoorbeeld "HartigeHapDbErco". Maak ook een database gebruiker aan met alle globale priveleges. Creëer m.b.v. een sql script de benodigde database tabellen en database data.

Het creëren van een database en een database user, lokaal op server, kan met <http://145.48.6.147/phpmyadmin/> m.b.v. user/pass is root/10ec4u

* Benader de database vanuit de broncode met jdbc:mysql://localhost/library". De localhost zorgt ervoor dat de applicatie op je laptop naar de database op je laptop gaat en op de Avans server naar de database op de Avans server.

Om je applicatie binnen Netbeans te kunnen blijven gebruiken, kies in Netbeans “File/Open Project” en ga naar de map <projectnaam>.In Eclipse kies je “Import …/Existing Maven projects”.

**Creëren van een Jenkins job voor jouw applicatie**

* Jenkins draait op <http://145.48.6.147:8081/>. Zonder in te loggen, kun je zien welke jobs er draaien. Om wijzigingen te kunnen maken, heb je user/passw idh10/10ec4u nodig.
* In Jenkins, creëer een nieuwe job en kies "maven 2/3 project"
* Vink aan “Discard Old Builds” en zet bij “Max # of builds to keep” 3 neer. Dit voorkomt dat de hard disk op de Avans serer vol raakt.
* Geef aan dat source code mgt via subversion of git gaat
* Geef de repository url van de online svn repo aan, bijvoorbeeld <https://subversion.assembla.com/svn/ercoargante/Library_IDH5/trunk>
* Geef aan waar de pom.xml staat
* Build triggers: poll scm H/5 \* \* \* \* (dit betekent dat Jenkins elke 5 minuten kijkt of er iets gecommit is)
* Als je geen blauw bolletje en zonnetje krijgt: klik op de job, klik op de mislukt bouwpoging, klik op "console output". Met die informatie kun je achterhalen wat er mis is.

**Sonar**

Het idee van een ontwikkelstraat is dat er meerdere stappen worden doorlopen. Dit is precies waar Jenkins voor bedoeld is. We zullen een extra stap toevoegen in de vorm van het tool Sonar, dat broncode analyse doet.

* In de Jenkins job die je hebt gemaakt, druk op “Post-build actions” (en dus niet “Post Steps”).
* Kies Sonar en kies “Sonar JavaSE”
* De volgende keer dat de Hartige Hap gebouwd wordt in Jenkins, als gevolg van een SVN commit, zal ook een Sonar analyse worden gedaan. Je kunt ook een build forceren met “build now”.
* De resultaten van de Sonar analyse zijn te vinden op <http://145.48.6.147:9000/>.

**Achtergrond: Jenkins security aspecten van de huidige setup**

Je zou kunnen denken dat Jenkins een zwakke schakel vormt in de security van het geheel. Jenkins kan immers bij alle SVN en GIT repo’s. Meestal wordt Jenkins gebruikt op een intranet waardoor security niet (of in mindere mate) door Jenkins zelf geregeld hoeft te worden.

* De rest van de wereld kan de Jenkins jobs zien die draaien en ook de details zoals de broncode
* Alle studenten kunnen (elkaars) jobs wijzigen
* Bij het creëren van een job geef je svn credentials op. Jenkins is zodanig ontworpen dat de credentials niet te gebruiken zijn om bijvoorbeeld broncode te wijzigen of te wissen