



| | | |
|-----------------|------|---|
| WERKGROEP GAP | | / |
| 26 januari 2012 | Not. | / |

Aanwezig: Koen Meersman, Johan Vervloet

Over deze vergadering

Op de 'prevergadering' van de werkgroep informatica, hebben we eens samengezeten rond workflow voor GAP-ontwikkeling en rond het opzetten van een testomgeving.

Voorstel syteem workflow

GAP is een vrij specifiek project, omdat we met vrijwilligers werken. We zitten meestal op verschillende plaatsen, en iet iedereen heeft even veel tijd om aan het programma te werken. Alleen Johan heeft er structureel tijd voor. En de momenten dat we allemaal samen kunnen vergaderen of programmeren zijn schaars.

We stellen het volgende systeem voor (gebaseerd op git):

Op de server is er één main repository ('upstream'), maar daarnaast ook nog een eigen (remote) repository voor iedere developer. En natuurlijk heeft iedere developer ook ergens één of meerdere lokale working copies. Elke developer heeft read access op alle repositories, maar enkel write acces op zijn eigen repository.

Een developer houdt zijn working copy up to date door geregeld de changes uit de main repository in zijn working copy te mergen. Verder ontwikkelt hij op zijn working copy. Hij kan lokaal changesets committen, en die changesets kan hij ook pushen naar zijn eigen remote repository.

Als de developer vindt dat zijn wijzigingen 'klaar' zijn, en geïntegreerd mogen worden in de main repostory, moet hij ervoor zorgen dat zijn changesets gepusht zijn naar zijn remote repository, en stuurt hij naar de project maintainer een 'pull request'. De project maintainer kan dan dan de gevraagde wijzigingen lokaal mergen, bekijken en eventueel aanpassen. Pas als de project manager tevreden is over de wijzigingen, zal hij ze pushen naar de main repository.

Meer uitleg hierover: <http://progit.org/book/ch5-1.html> (onder het titeltje 'Integration-Manager Workflow').

Dit systeem wordt ondersteund door github. Op <http://help.github.com/fork-a-repo/> staat uitgelegd hoe alles praktisch in zijn werk gaat.

Ik (Johan) heb er zelf eens mee geëxperimenteerd, door wat taalfouten uit de Nederlandse vertaling van het betreffende tekstje uit het boek (pro git) te verbeteren. In praktijk:

De repository van 'pro git' staat hier: <https://github.com/progit/progit>

Mijn kloon van de repository is deze: <https://github.com/johanv/progit>

Ik commite een changeset naar mijn remote repository:

<https://github.com/johanv/progit/commit/f74cc96642b97cc641e38fc194ae47fe40d36f4d>

En ik stuurde een pull request naar de 'upstream': <https://github.com/progit/progit/pull/185>

We zullen zien of mijn changes accepteren :-)

Testomgeving

We bekeken ook hoe we een testomgeving kunnen opzetten, met automatische builds. Dit is het voorstel:

devsrv1

De GAP-liveomgeving gebruikt 2 servers:

- gapsrv1.chiro.wereld (webserver voor GAP frontend en backend, services voor synchronisatie van en naar Kipadmin)
- kipsrv1.chiro.lokaal (sql server 2005 database, met databases voor kipadmin en GAP)

De testomgeving zou volledig weg moeten van de twee bovenvermelde server. We hebben een windows-2008 server ter beschikking voor ontwikkeling (devsrv1), en we stellen voor om alles van de testomgeving daar te draaien:

- webserver voor frontend en backend GAP
- databases voor GAP en Kipadmin
- synchronisatieservices tussen GAP en Kipadmin

Om dezelfde situatie te simuleren als de live-omgeving, zullen we devsrv1 verhuizen naar het chiro.wereld-domein. Op devsrv1 zullen we dan SQL Server draaien met een service account uit chiro.lokaal. Op die manier reproduceren we de problemen die we ondervinden voor de communicatie GAP-Kipadmin over message queues.

(<https://develop.chiro.be/trac/cg2/ticket/927>).

We zullen op devsrv1 een instantie van SQL server 2010 installeren, want op termijn gaat de migratie van de live-omgeving naar 2010 toch eens moeten gebeuren. Op die manier kunnen we al eens bekijken hoe gemakkelijk/moeilijk die migratie zal zijn, en of er zich problemen voordoen.

In praktijk zijn er wel wat zaken die afhangen van onze databaseserver (zie <https://adminwiki.chiro.lokaal/doku.php/devel:dbprocessenservices>). Die gaan we op dit moment niet allemaal op de testomgeving laten draaien. Enkel de testomgeving voor GAP zal naar devsrv1 verhuizen, samen met de synchronisatie GAP-Kip. Ook zullen we ergens op een (virtuele?) PC een Kipadmin-client opzetten die connecteert met de testdatabase op devsrv1.

Om te vermijden dat er gegevens van de test in de live terechtkomen, blokkeren we alle verkeer tussen devsrv1 enerzijds en gapsrv1 en kipsrv1 anderzijds; vermoedelijk lukt dat wel via de windows firewall op devsrv1.

Koen krijgt lokale adminrechten op devsrv1 en sysadminrechten op de databaseserver daarop, om mee te kunnen helpen aan de testomgeving.

database setup

Op devsrv1 zetten we een nieuwe database-instantie op: DEVSRV1\sql2010

Met daarop deze databases:

- gap-test
- kipadmin-test
- gap-dev
- kipadmin-dev

gap-test en kipadmin-test zullen de databases zijn voor de testomgeving, die we via een script automatisch ophalen van de laatste backup van de live-omgeving. Dat script zal de gegevens daarna 'randomizen', om privacyproblemen te vermijden. Gap-dev en kipadmin-dev zullen de databases zijn waarop ontwikkeld wordt. (ter vervanging van de ontwikkelingsdatabases die nu op DEVSRV1\sql2k5 draaien). In het ideale geval hernoemen we de livedatabases naar gap-live en kipadmin-live, kwestie van de kans op vergissingen zo klein mogelijk te houden.

automatische builds

Op regelmatige basis (bijv. iedere nacht), gebeurt onderstaande procedure automatisch:

- de testomgeving wordt stilgelegd
- de recentst beschikbare full backup van de liveomgeving wordt teruggezet naar de testomgeving
- het databasemigratiescript wordt gerund¹. Errors en warnings worden gerapporteerd.
- de data in de testomgeving wordt 'gerandomized' (namen, adressen,... door elkaar gegoooid, e-mailadressen en telefoonnummers vervangen door iets willekeurig)
- de recentste source wordt gedownload gecompileerd. Errors en warnings worden gerapporteerd.
- de nieuwe versie wordt geïnstalleerd, de configuratiebestanden worden vervangen door de configuratiebestanden van de live-omgeving.²
- de live-omgeving wordt opnieuw gestart.

¹Alle wijzigingen die aan de database moeten gebeuren sinds de laatste release worden verzameld in 1 script, dat uitgevoerd moet worden om de structuur van de livedatabase om te zetten naar de structuur van de devdatabase.

²Na te kijken of alle configuratie wel via de configuratiebestanden loopt, en er niets van configuratie mee in exe's of dll's wordt gezet.