Git(hub)

unset(svn);
namespace github;
use git as vcs;

Introductie

Wanneer men werkt met code - ongeacht php, javascript, ruby, html etc - men loopt er tegenaan dat hoe groter het project of met hoe meer mensen men samenwerkt, hoe meer verschillende taken men tegelijkertijd oppakt, een versiebeheersysteem onmisbaar wordt.

HostingXS stapt na lang subversion gebruikt te hebben, over op git en de interface github. Deze keuze is cruciaal in de jarenlange development die dit bedrijf heeft uitgevoerd. Om duidelijkheid te geven wat versiebeheer is en waarom deze onmisbaar is geworden in een bedrijf dat intern meerdere applicaties ontwikkeld, is dit document ontstaan.

hostingxs

Wat is versiebeheer

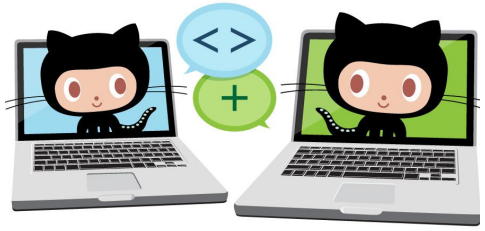
Versiebeheersystemen houden wijzigingen bij in een set bestanden. De set wordt een "**repository**" (afgekort repo) genoemd wanneer deze onder versiebeheer zit. Iedere wijziging in de set wordt gekoppeld aan een gebruiker, een tijdstip en de informatie van wat er exact gewijzigd is. De verandering in de set wordt een "**commit**" genoemd. Door deze informatie kan er altijd teruggegrepen worden op een oudere versie; een fout maken is dus niet direct een ramp.

Wanneer meerdere ontwikkelaars tegelijkertijd werken aan dezelfde set - repo - kunnen ze elkaars wijzigingen volgen en in hun eigen ontwikkelomgeving samenvoegen, dit wordt "**mergen**" genoemd. Moderne versiebeheersystemen kunnen automatisch de wijzigingen van commits samenvoegen. Toch kan het voorkomen dat er reeds een wijziging is uitgevoerd op regels waar jij ook aan bezig was; bij het samenvoegen kan dan een "**conflict**" ontstaan. Het versiebeheersysteem geeft je daarmee de melding dat er gekozen moet worden welke regels de beste zijn, dit kiezen heet "**resolven**". Het versiesysteem zorgt er dus voor dat je niet zomaar andermans werk teniet doet.

Doordat gebruikers van de set verschillende takken ("**branches**" genoemd) kunnen maken, is het mogelijk om de ontwikkeling van een bepaalde feature te scheiden van de draaiende (live) website. De branches kunnen naderhand weer met de live branch samengevoegd - gemerged - worden.

Meer weten?

- Een doorzoekbare geschiedenis van wijzigingen
- Wijzigingen door meerdere gebruikers tegelijkertijd
- Meerdere versies naast elkaar
- Voorgaande en klaargezette wijzigingen in 1 oogopslag inzien
- Verschillen tussen wijzigingen bekijken
- Redenen van wijzigingen bijhouden



Subversion (svn) versus Git

Git stapt af van een centrale repository waarmee verbonden moet worden. De gebruiker bouwt zijn repository lokaal (op de computer) op en commit wat nodig is naar een centrale repository. Formeel is SVN een centraal versiebeheersysteem en git een gedistribueerde.

In SVN is er meer kans op fouten bij het mergen. Met name wanneer de hoeveelheid merges toenemen vanuit meerdere branches. Het aantal conflicts kan daarbij flink escaleren. Zover dat je meer tijd kwijt bent het conflict resolven dan met het committen.

Met git kan je eenvoudiger overstappen naar een andere branch; commit de lopende wijzigingen naar een nieuwe branch waarna je eenvoudig kan switchen.

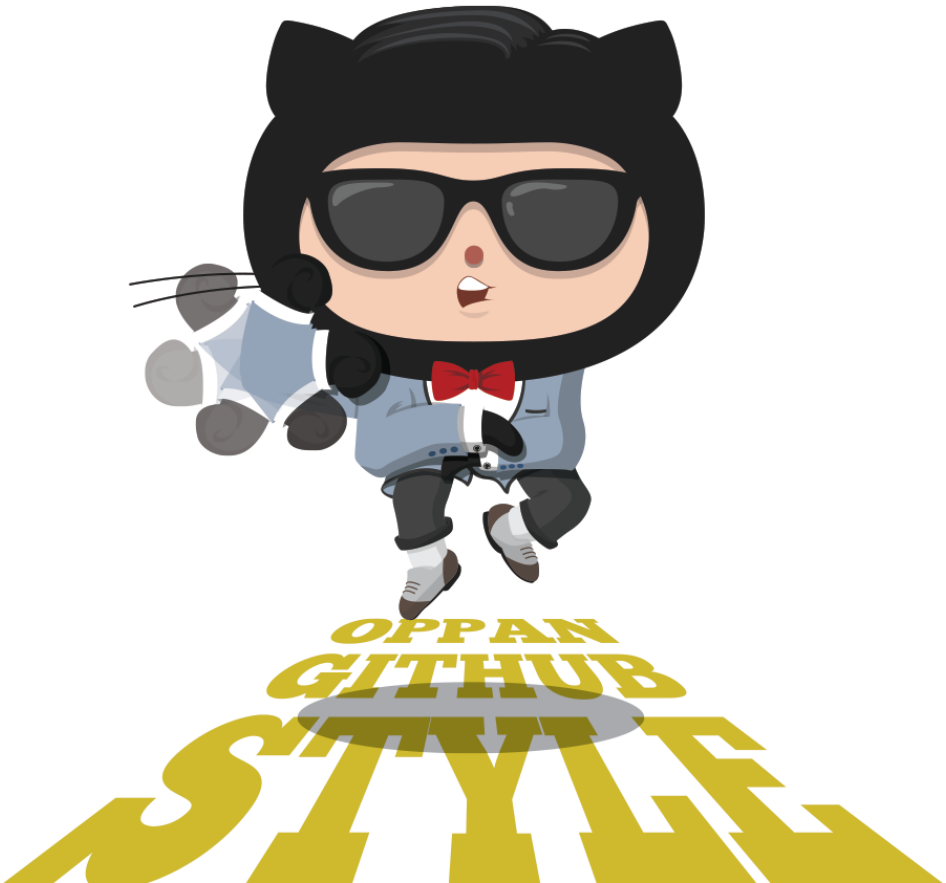
Met git ben je niet afhankelijk van de beschikbaarheid van je online repository.

Vergelijking door een gebruiker (engels)

Github, de git interface

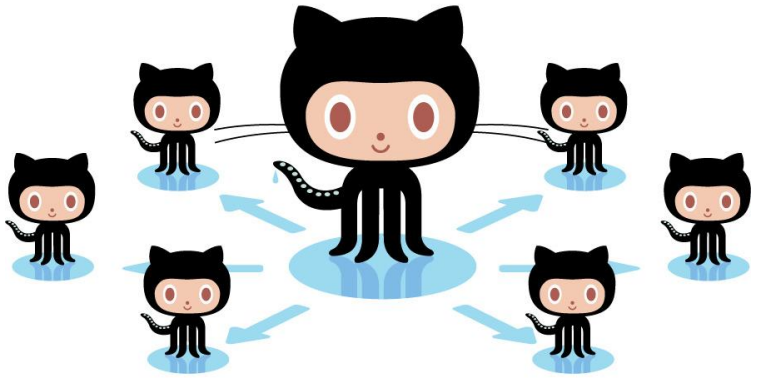
Github is de heilige koe van de git interfaces. Het biedt veruit de meeste features en de afgeschermde repositories zijn niet eens zo heel duur (5 voor 7 dollar). Maar wat is er dan zo brilliant aan:

- Helpt je in iedere stap!
- Open source (publieke) repositories zijn altijd gratis.
- Alle commits, branches en code online inzien en vergelijken. Commits online mergen naar willekeurig welke branch. Via een online editor de code direct aanpassen en committen.
- Geintegreerd ticketsysteem (zoals trac); koppel commits aan tickets zodat ze direct opgelost worden.
- Geintegreerd wikisysteem om informatie over je code te onthouden.
- Statistieken van je repository inzien; punchcards, commits etc.
- Online snippetsysteem genaamd gist; schrijf je korte stukjes publieke of privé code.
- Integreer andermans publieke repo's in jouw code of dupliceer een versie voor jezelf.
- Geweldige community en een uitgebreid leersysteem (de codeschool les bijvoorbeeld).
- Koppeling met een lange lijst aan andere applicaties.
- Scheiding van organisaties en gebruikers.
- Code highlighting, emotes, inline code commentering



Terminologie

- repo(sitory)
  de set bestanden toebehorend aan een project
- commit
  een (set) veranderingen die naar de server verzonden zijn om de repo te updaten
- forken
  het dupliceren van andermans repo voor eigen gebruik of om de code uit te breiden
- branch
  een tak van de repository, waarin andere wijzigingen worden uitgevoerd
- merge(n)
  een samenvoeging van branches naar een branch, hierdoor worden branches gelijkgetrokken
- pull request
  een verzoek indienen aan de eigenaar van de repo om een fork of branch te mergen naar de repo



hostingxs

github.com/HostingXS

Do you Git it?

Leer zelf de basis via de interactieve les:

try.github.com

Volledige git documentatie:

git-scm.com

Branching en merging:

git-scm.com

Github hulp en documentatie:

help.github.com

Verschil in user en organisatie rechten:

help.github.com

hostingxs