**Leden**: Wouter Bollaert

Maarten Hoste

Stijn Kuppens

Tim Leenaers

Marco Tordoor

|  |  |
| --- | --- |
| **Academiejaar**: 2012 – 2013 | **Opleidingsonderdeel**:  Geïntegreerd softwareproject |
| **Opleiding**: Toegepaste Informatica | **Jaar**: 2 |
| **Groep**: 9 |  |

**HANDLEIDING VERSIEBEHEER**

**Professoren**: Jan Willekens

Kerstin Nys

Erik Ulrichts

Marijke Sporen

# Wat is versiebeheer?

Versiebeheer is een systeem dat veranderingen in een bestand bijhoudt. Hierdoor kan je later specifieke versies opvragen.

# Waarom gebruik maken van een versiebeheersysteem?

Als je een website ontwerpt en je wilt elke versie van een afbeelding of opmaak bewaren, is het niet onverstandig om een versiebeheersysteem te gebruiken. Op die manier kan je eerdere versies van bestanden terughalen, veranderingen bekijken, zien wie het bestand voor het laatst aangepast heeft en nog veel meer. Door het gebruik van een versiebeheersysteem kan je dus gemakkelijk een situatie terugdraaien als je een fout maakt of een bestand kwijtraakt. Dit neemt weinig tijd in beslag.

# Hoe gebruik je een versiebeheersysteem?

Er zijn verschillende soorten van versiebeheersystemen:

* Lokale versiebeheersystemen
* Gecentraliseerde versiebeheersystemen
* Gedistribueerde versiebeheersystemen

## Lokaal versiebeheersysteem

Veel mensen hebben hun eigen manier van werken met een versiesysteembeheer. Ze kopiëren bestanden simpelweg naar een andere map. Deze methode is erg gemakkelijk en wordt daarom veel gebruikt. Het is echter ook gemakkelijk om naar een verkeerd bestand te schrijven, over bestanden heen te kopiëren, enzovoort…

Om dit probleem op te lossen hebben programmeurs vroeger lokale versiebeheersystemen ontwikkeld die een database hadden om veranderingen te beheren.

Een lokaal versiebeheersysteem gebruiken we echter best niet voor ons project, omdat we met vijf personen tegelijk werken aan hetzelfde project.

## Gecentraliseerd versiebeheersysteem

Zoals reeds gezegd is het bij lokale versiebeheersystemen een probleem om samen te werken op verschillende computers. Om dit probleem op te lossen werd een gecentraliseerd versiebeheersysteem ontwikkeld. Deze systemen hebben één centrale server waarop alle versies van elk bestand staan. Dit was lang de standaard voor versiebeheer.

Deze manier van versiebeheer biedt veel voordelen ten opzichte van een lokaal versiebeheersysteem. Iedereen weet bijvoorbeeld wat andere groepsleden op dat moment aan het doen zijn. Beheerders hebben ook volledige controle over wie welke bestanden kan bekijken of aanpassen.

## Gedistribueerd versiebeheersysteem

In een gedistribueerd versiebeheersysteem downloaden gebruikers niet alleen de laatste versies van de bestanden, maar de hele geschiedenis wordt gekopieerd. Als een server dan neergaat, kan de geschiedenis van elke gebruiker gekopieerd worden naar de server om deze te herstellen.

# GIT

GIT is niet zoals andere centrale versiebeheersystemen, bijvoorbeeld ‘Subversion’. Bij centrale versiebeheersystemen worden revisies standaard geüpload naar een server wanneer een nieuwe versie ingediend wordt door een gebruiker.

Bij GIT daarentegen worden versies bewaard als er veranderingen plaatsvinden. De lokale opslagplaats kan op elk moment gesynchroniseerd worden met de server. Hier is geen internetverbinding voor nodig, want versies kunnen op elk moment ingediend worden. Bij centrale versiebeheersystemen is dit niet het geval.

Er is wel een nadeel. Gebruikers kunnen lokaal aan een versie werken en deze op hetzelfde moment synchroniseren met de server. Hierdoor kunnen conflicten bestaan, aangezien de gebruikers niet dezelfde aanpassingen hebben gemaakt. Als een gebruiker zijn versie synchroniseert met de server, zal hij conflicten moeten oplossen. Hiermee wordt echter rekening gehouden in de Git-software.

# Hoe installeer je GIT?

Volg volgende stappen:

* Download de laatste versie van GIT
* Installeer de gedownloade versie
* Configureer je GIT-instellingen (via de GIT-Bash)
* Gebruikersnaam
* Email
* Wachtwoord