

ICT-Empowerment Junior App Tester

Versiebeheer met git en github



Samen op weg naar jouw succes!

© 2019 OTech BV - niets uit deze uitgave mag worden vermenigvuldigd en/of verspreid - zonder voorafgaande, schriftelijke toestemming van de auteur

Zonder versiebeheer

Samen aan bestanden werken in een gedeelde map:

- “Ik werk nu aan bestand XYZ!”
(dus blijf er met je vingers vanaf)
 - moeilijk als je niet op dezelfde locatie werkt
 - hoe weet je wanneer iemand weer klaar is met dat bestand?
- “20180330-RS-ontwerp-v23-concept.docx”
 - onleesbaar en moeilijk onderhoudbaar
 - hoe voeg je wijzigingen weer samen?

Foutgevoelig!



Wat is versiebeheer

Een systeem dat...

- ...je wijzigingsgeschiedenis bijhoudt
- ...bijhoudt wie wanneer wat heeft aangepast
- ...het mogelijk maakt om naar een eerdere toestand terug te gaan

...zodat je veilig samen aan bestanden kunt werken.



Git

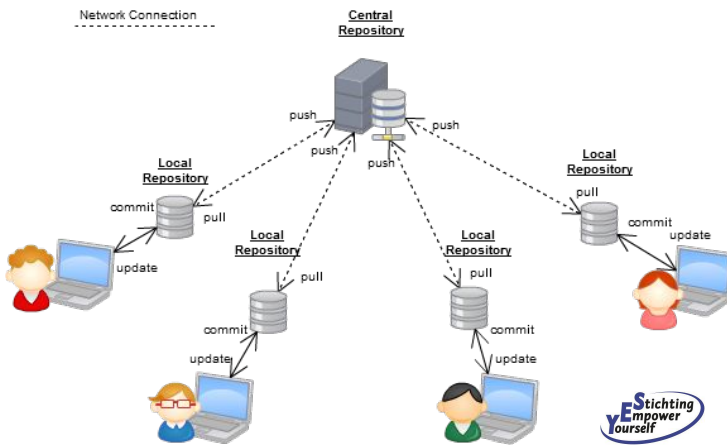
- distributed version control
- gebruikers hebben eigen kopie
- je werkt lokaal aan je bestanden



Git terminologie

- snapshot: recorded state of all files at a given point in time
- commit: the act of creating a snapshot
- repository: a collection of snapshots
- cloning: duplicating a repository (e.g. from central to local)
- pulling: updating your local repository
- pushing: updating the central repository





Git: installeren

- Download van: <https://git-scm.com/>
- Installeer en start `git bash`
- Configureer:


```
git config --global user.name "Your Name"
git config --global user.email your.email@example.com
```

Git: start een nieuw project

```
mkdir MyNewProject
cd MyNewProject
git init
```

Git: lokaal werken aan een project

- bestanden 'stagen' (toevoegen aan de index):
`git add <filename>`
- bestanden toevoegen aan local repository:
`git commit`

Waar zijn mijn bestanden?



Github

- service that provides storage for central repositories
- extra services on top of git, e.g.:
 - wiki
 - issue management
 - pull requests



Git: werken met een remote repository

- Verder met een bestaand project:
`git clone <URI>`
- Voeg een remote toe:
`git remote add origin <server>`
- bestanden ophalen van centrale server:
`git pull`
- bestanden toevoegen aan remote repository
`git push <remote> <branch>`



Git: informatie opvragen

- `git status`
- `git log`
- `git diff`
- `git gui`



Git: tags

Semantic Versioning 2.0.0 (<https://semver.org/>)

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.



Version control best practices

- if it isn't in version control, it doesn't exist
- commit often (keep changes small)
- do *not* commit generated artifacts
- build and test your changes before committing
- never work on the master branch
- clearly describe your changes

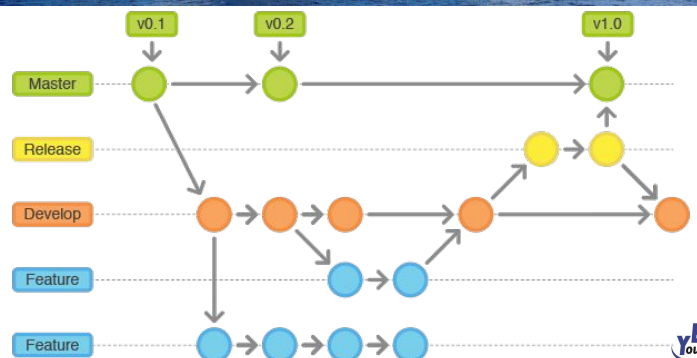


In case of fire 

> `git commit`
> `git push`
> `git outofhere`



Git branching





Bronnen

<https://try.github.io/>

<https://git-scm.com/book/nl/v2>

<http://rogerdudler.github.io/git-guide>

<http://marklodato.github.io/visual-git-guide>

<https://www.learnenough.com/git-tutorial>

