howest
hogeschool

Branches

Continuous Integration Basics

Wat zijn branches?

Onafhankelijk ontwikkelingspad (aftakking) in version history

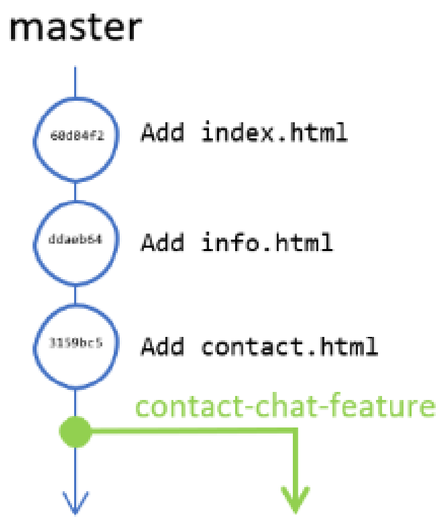
- Tijdelijk **experimenteren** zonder originele repo te “bevuilen”
- **Teamwork**: onafhankelijk aan bv. **feature** werken zonder anderen te verstoren
- **Stabiele versie** (productie, demo, …) niet verstoren tot feature klaar is
- Je kan altijd **terugkeren** naar aftakkingspunt!
- Indien tevreden over werk: **samenvoegen *(= merge)*** van branches

Branches: demo

Startrepo via opdracht op Leho (instructies in README)

Commando	Beschrijving
git branch	Toon overzicht van de branches (en actieve branch)
git branch *my-branch*	Maak branch aan met naam *my-branch* (afgetakt van actieve branch)
git checkout *my-branch*	Wissel naar branch met naam *my-branch*
git checkout -b *my-branch*	Maak branch met naam *my-branch* aan en check deze meteen uit
git merge *some-branch*	Merge de branch *some-branch* **in de actieve branch**
git push --set-upstream origin *my-branch* ***Korte versie:*** git push -u origin *my-branch*	Push je **lokale actieve** branch naar een **nieuwe remote** branch *my-branch.* Na deze initiële push kan je gewoon met “git push” werken voor alle volgende commits.

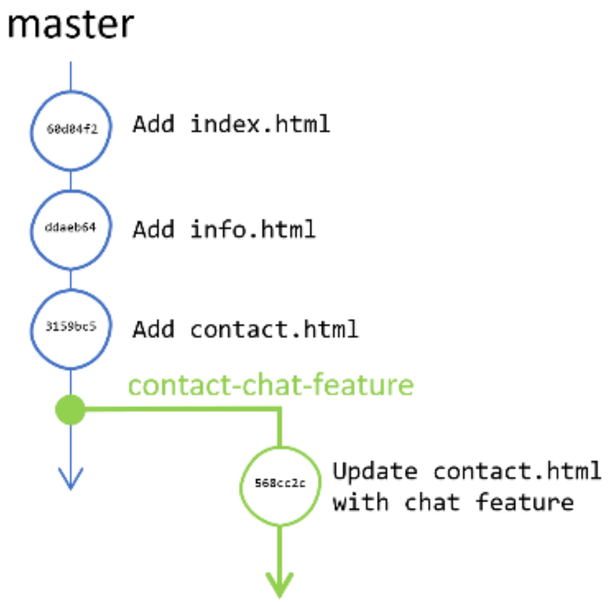
Branches visueel




```
ConsoleFriend@Desktop ~/Desktop/BranchDemo (master)
$ git branch contact-chat-feature

ConsoleFriend@Desktop ~/Desktop/BranchDemo (master)
$ git branch
  contact-chat-feature
* master
```

```
ConsoleFriend@Desktop ~/Desktop/BranchDemo (master)
$ git checkout contact-chat-feature
Switched to branch 'contact-chat-feature'

ConsoleFriend@Desktop ~/Desktop/BranchDemo (contact-chat-feature)
$ git branch
* contact-chat-feature
  master
```

Branches visueel



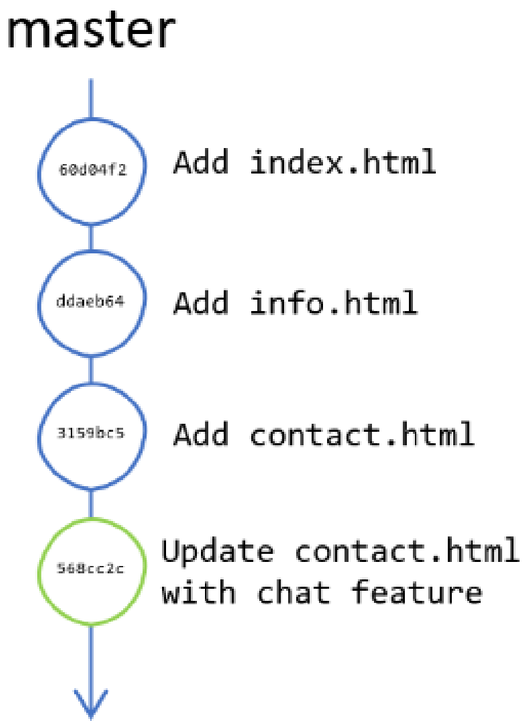

```
ConsoleFriend@Desktop ~/Desktop/BranchDemo (contact-chat-feature)
$ code contact.html

ConsoleFriend@Desktop ~/Desktop/BranchDemo (contact-chat-feature)
$ git add contact.html

ConsoleFriend@Desktop ~/Desktop/BranchDemo (contact-chat-feature)
$ git commit -m "Update contact.html with chat feature"
[contact-chat-feature 568cc2c] Update contact.html with chat feature
 1 file changed, 2 insertions(+)

ConsoleFriend@Desktop ~/Desktop/BranchDemo (contact-chat-feature)
$ git log --oneline
568cc2c (HEAD -> contact-chat-feature) Update contact.html with chat feature
3159bc5 (master) Add contact.html
ddaeb64 Add info.html
60d04f2 Add index.html
```

Branches visueel


```
ConsoleFriend@Desktop ~/Desktop/BranchDemo (contact-chat-feature)
$ git checkout master
Switched to branch 'master'

ConsoleFriend@Desktop ~/Desktop/BranchDemo (master)
$ git log --oneline
3159bc5 (HEAD -> master) Add contact.html
ddaeb64 Add info.html
60d04f2 Add index.html
```

Laatste commit **nog niet** aanwezig op master branch!

Branches visueel



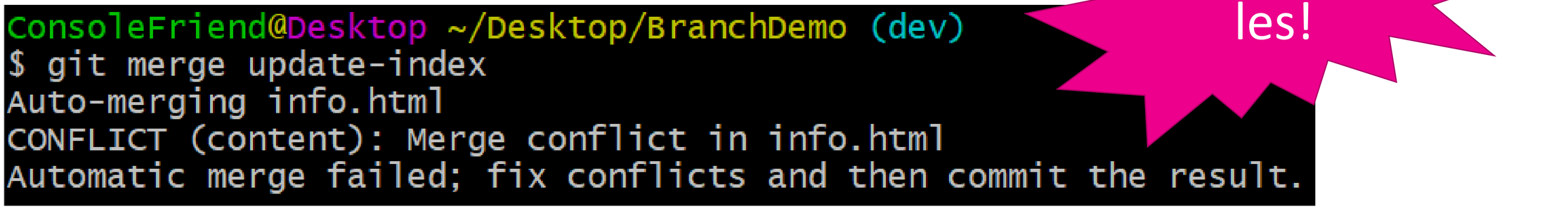

```
ConsoleFriend@Desktop ~/Desktop/BranchDemo (master)
$ git merge contact-chat-feature
Updating 3159bc5..568cc2c
Fast-forward
 contact.html | 2 ++
 1 file changed, 2 insertions(+)

ConsoleFriend@Desktop ~/Desktop/BranchDemo (master)
$ git log --oneline
568cc2c (HEAD -> master, contact-chat-feature) Update contact.html with chat feature
3159bc5 Add contact.html
ddaeb64 Add info.html
60d04f2 Add index.html
```

Na merge komt de commit ook op master

Merge conflicts

Wat als je op meerdere branches **dezelfde bestanden** aanpast?

- Mergen kan **conflicten** opleveren
- Conflict = **andere** aanpassing van **zelfde stuk code** in **zelfde bestand**
- Conflicten moeten **manueel opgelost** worden!
  - Git weet niet wat het gewenste eindresultaat is…



Volgende les!

howest hogeschool

Conventies branches: naamgeving

Naamgeving branches:

- kebab-case
- **duidelijke** en passende omschrijving!

✔	✘
login-screen	MyNewFeature
attack-animations	attackAnimations
refactor-email-service	refactorEmailService

howest hogeschool

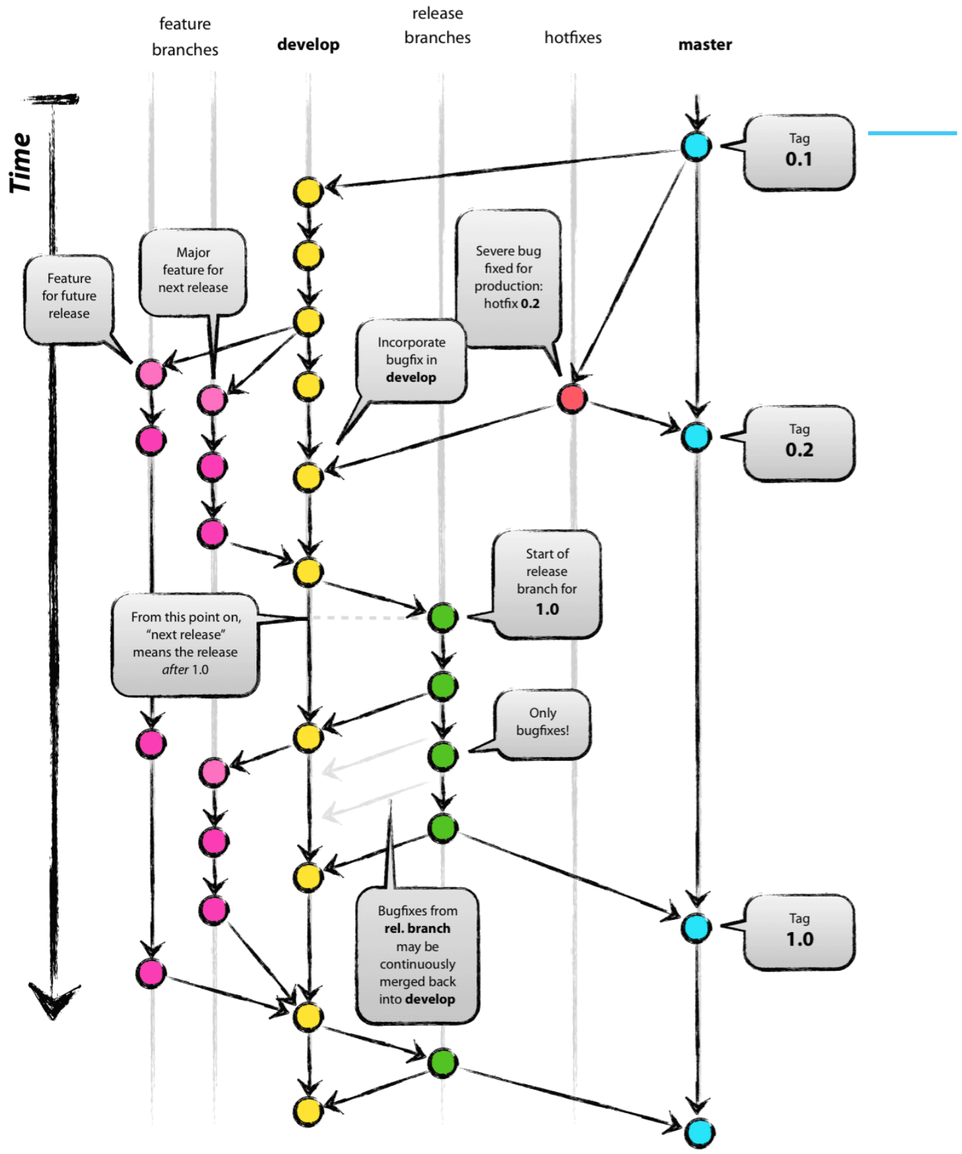
Conventies branches: gitflow

In de toekomst zal je met *heel wat* branches werken!

Komt verder aan bod doorheen je opleiding.

We starten met:

- master/main
- dev (of develop)
- feature branches



howest hogeschool

Conventies branches: eenvoudige start

Branch	Inhoud
`master / main`	**Finale** versie (productie-versie) = *default branch*
`dev / develop`	Branch met nieuwe **ontwikkelingen**: • Afgetakt van `master/main` • Aftakpunt voor feature branches • Feature branches worden in `dev` gemerged • Van `dev` naar `master/main` als feature(s) helemaal afgerond en uitgebreid getest

Feature branches	• Afgetakt van `dev`
	• Nieuwe branch per feature/wijziging/…
	• **Naamgeving**: feature/xyz
	  • feature/undo-redo
	  • feature/reset-password

`Master` of `main`? Conventie was `master`—GitHub recent gewijzigd naar `main`

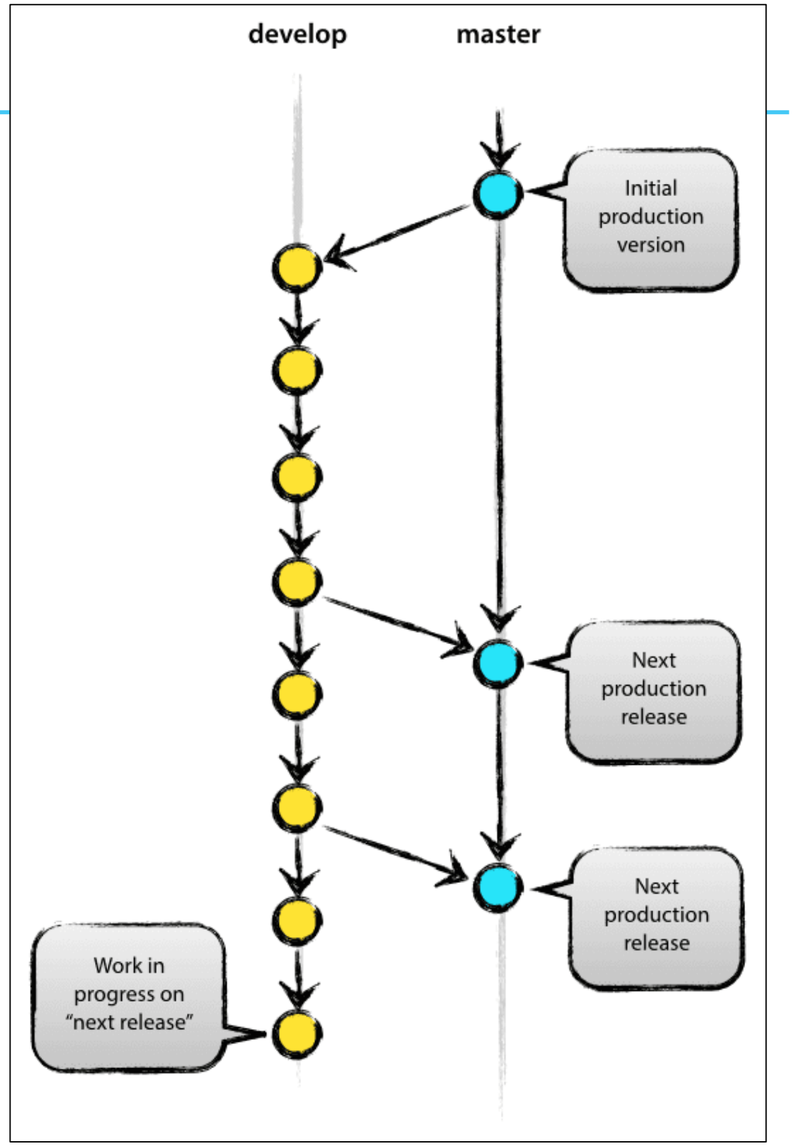
De twee hoofdbranches: master en dev

develop = work IN PROGRESS
→ Voor volgende release

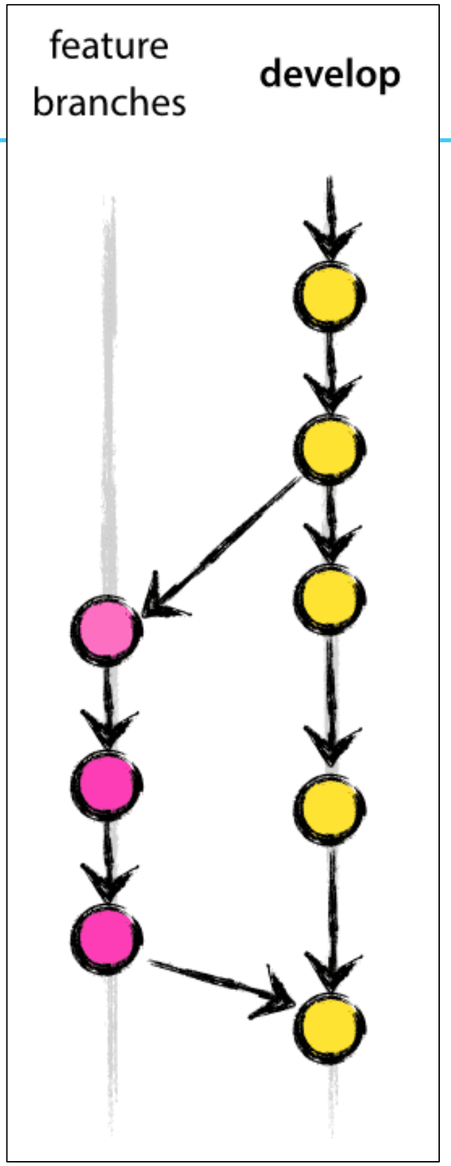
master = huidige STABIELE productie versie

Nooit rechtstreeks op **master** werken!

master en **dev** blijven altijd bestaan (onbeperkte levensduur)



Feature branches



Tijdelijke aftakking van **develop** voor:

- Ontwikkeling feature/improvement/fix
- Experimentje (dat misschien nooit gemerged wordt)
- …

Vaak **meerdere parallele** feature branches in gebruik!

- Teamwork
- Meerdere teams op shared code base

Afgetakt van	Gemerged naar
develop	develop

Feature branches: naamgeving

Naamgeving feature branches:

- begint met prefix **feature/**
- kebab-case
- alles kan, behalve master, dev, release- of hotfix-
- **duidelijke** en passende omschrijving!

✔	✕
`feature/login-screen`	`features/login-screen`
`feature/attack-animations`	`attack-animations`
`feature/refactor-email-service`	`Feature/refactorEmailService`

howest hogeschool

Commando’s feature branch

Aanmaken

```
$ git checkout -b feature/my-feature dev
```

Tak af van dev (ook als je momenteel op andere branch zit)

Afronden (mergen)

```
$ git checkout dev
$ git merge feature/my-feature
$ git push
```

Verwijderen

```
$ git branch -d feature/my-feature
$ git push origin --delete feature/my-feature
```

Lokaal

Remote

howest hogeschool

Release en hotfix branches

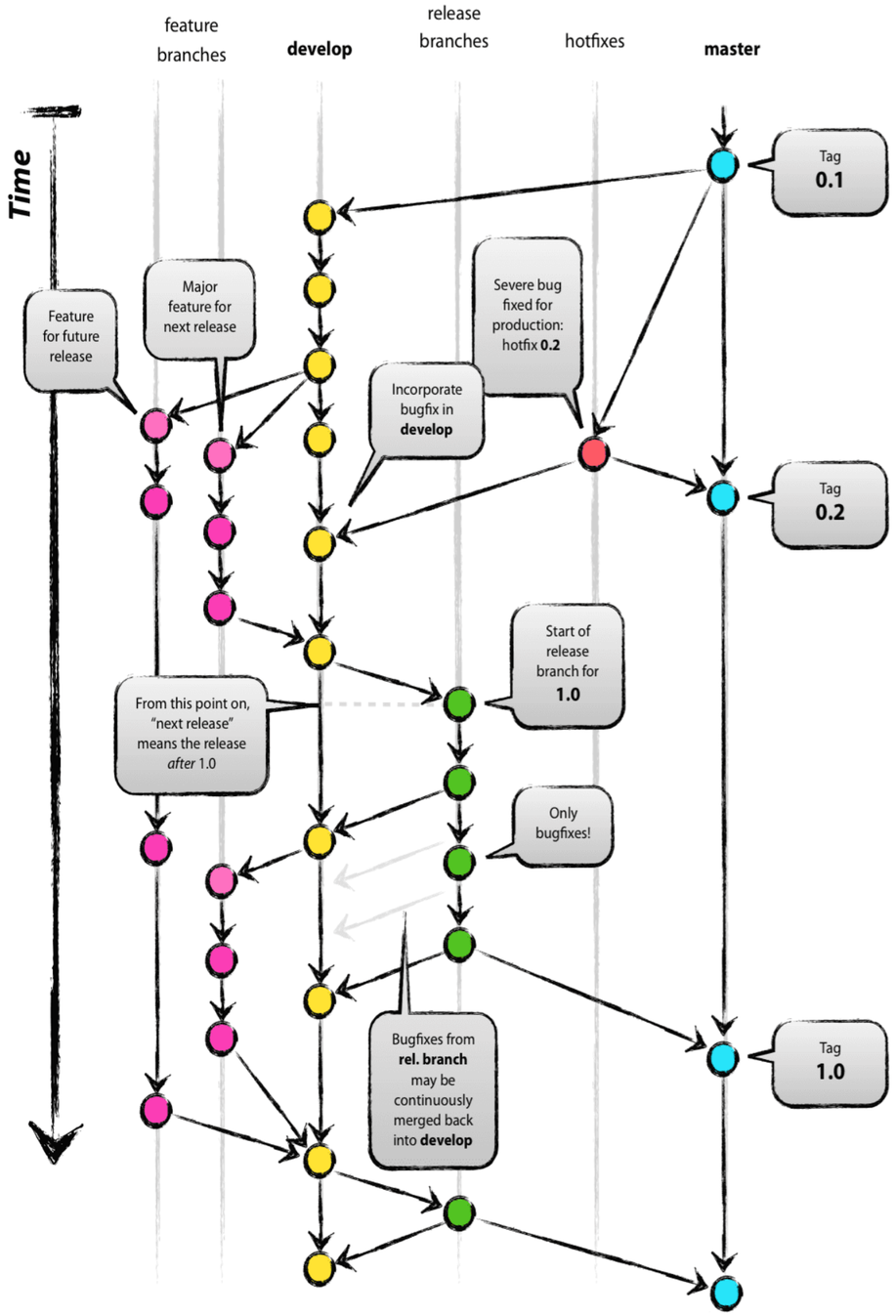
Zie cursus deel 3 §4 en §5

Release branches

- Tussenstap van dev naar master
- Testen, version bump, fixes, ...
- **Van** dev **naar** master & dev

Hotfix braches

- Kritieke fixes die snel naar productie moeten
- Zoveel mogelijk vermijden!
- **Van** master **naar** master & dev/release



howest hogeschool