Effektivt arbete med Git – Gitflow

MALMÖ UNIVERSITY

Gitflow: En överblick

En vanlig modell för att arbeta med Git (Driessen 2010)

- Välanvänt och váltestat
- Använder mycket branching
- Väl anpassat för tighta utvecklingsteam
- Inte lika väl anpassat för större, löst organiserade team
  - Varje utvecklare måste kunna göra en push

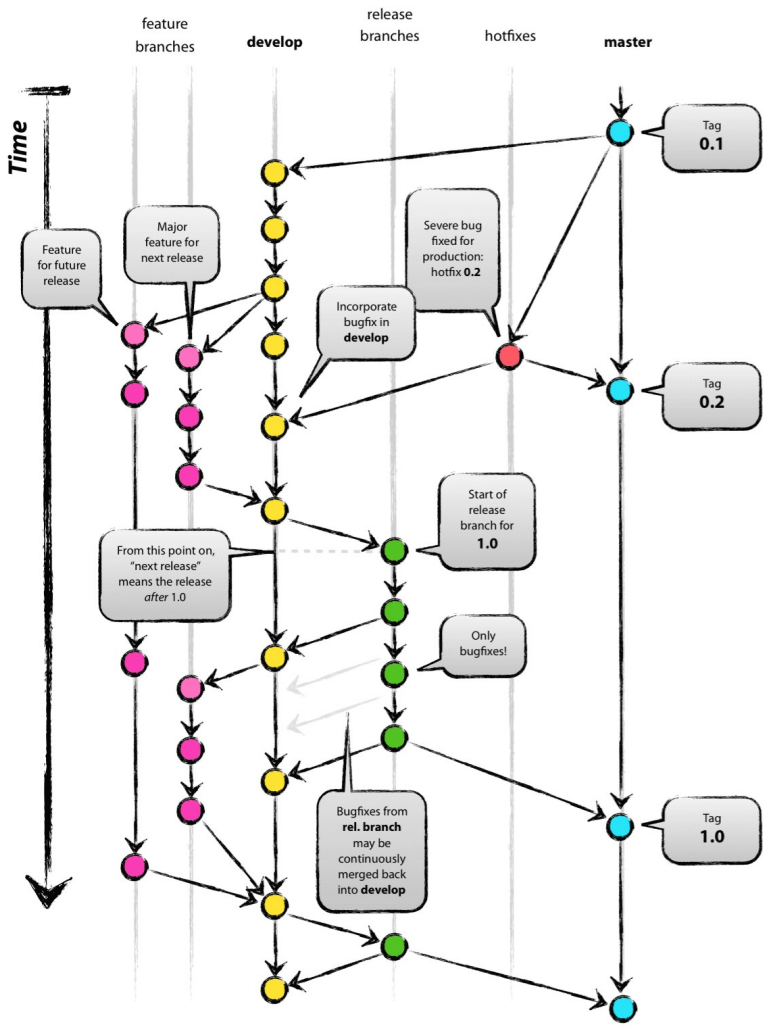
Driessen, Vincent. *A successful Git branching model*. 2010.
Source: https://nvie.com/posts/a-successful-git-branching-model/

MALMÖ UNIVERSITY

Gitflow: CLI-verktyget

Driessen skapade ett CLI-verktyget för att förenkla användandet av Gitflow, kallat *git-flow*

- Tillgängligt för Windows, macOS, Linux, etc.
- Underhålls inte längre
- Nuvarande “aktiva” version (en fork!): https://github.com/petervanderdoes/gitflow-avh

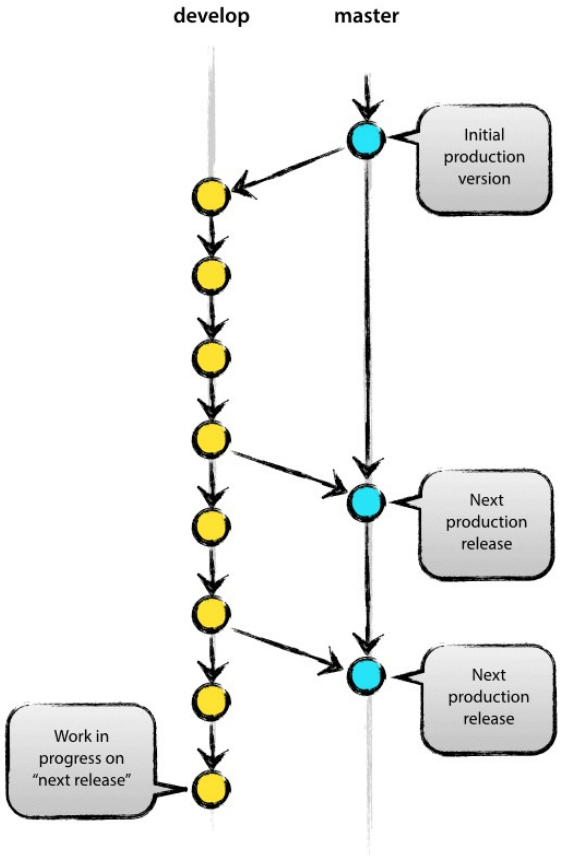


MALMÖ UNIVERSITY

master och *develop*

De två eviga avgreningarna

- *master*
  - Den “orörbara” grenen, här sker ingen utveckling
  - Fungerar som en samling av releaser
- *develop*
  - Den “smutsiga” grenen, all utveckling sker här
  - Avgrenad från den första commiten till *master*
  - Används som bas för (nästan) alla andra grenar



MALMÖ UNIVERSITY

master och *develop*

Initiera ett repository

- Med *git-flow*

```
git flow init
```

- Standard git

```
git init
git branch develop
git push –u origin develop
```

MALMÖ UNIVERSITY

Stödjande grenar

Alla andra grenar är tillfälliga:

- Uppmuntrar separation-of-concerns
- Förhindrar att för mycket görs i en gren
- Håller historiken hyfsat ren och förståbar

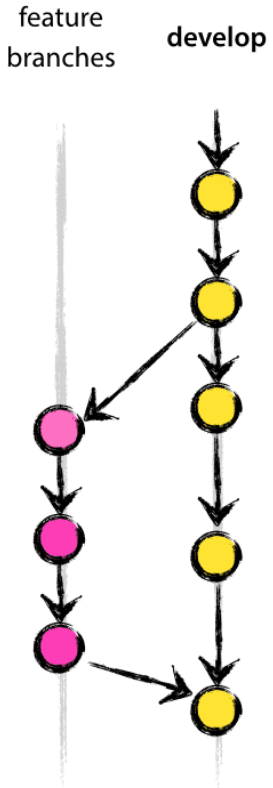
Viktigt! Alla grenar är tekniskt sett lika. Allt detta är bara mänskliga konventioner.

Feature branches

Används för ny, mer omfattande funktionalitet, även för framtida releaser

Viktiga punkter:

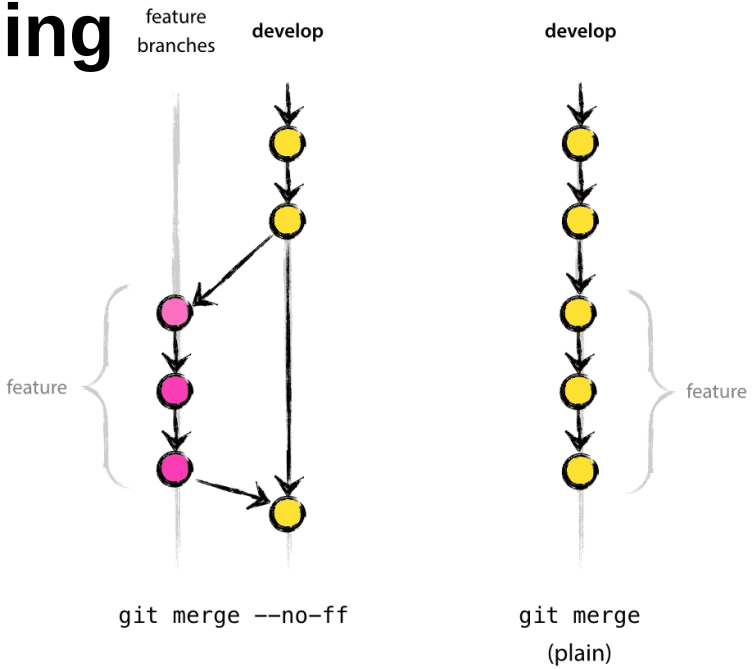
- Avgrenas alltid från *develop*
- Slås alltid samman med *develop*
- Kan heta vad som helst (Johan föreslår *feature/<description>*, t.ex. *feature/login*)



MALMÖ UNIVERSITY

Feature branches: merging

- *--no-ff* (“no fast forward”) behåller historik och trädets grenar
- Fast forward håller trädet tunt, men förlorar historisk mening
  - Svårare att följa förändringar
  - Svårare att gå tillbaka i historiken



MALMÖ UNIVERSITY

Feature branches

Skapa en feature-branch

- Med *git-flow*

```
git flow feature start branch_name
```

- Standard git

```
git checkout develop
git checkout -b branch_name
```

MALMÖ UNIVERSITY

Feature branches

Avsluta en feature-branch

- Med *git-flow*

```
git flow feature finish branch_name
```

- Standard git

```
git checkout develop
git merge --no-ff branch_name
```

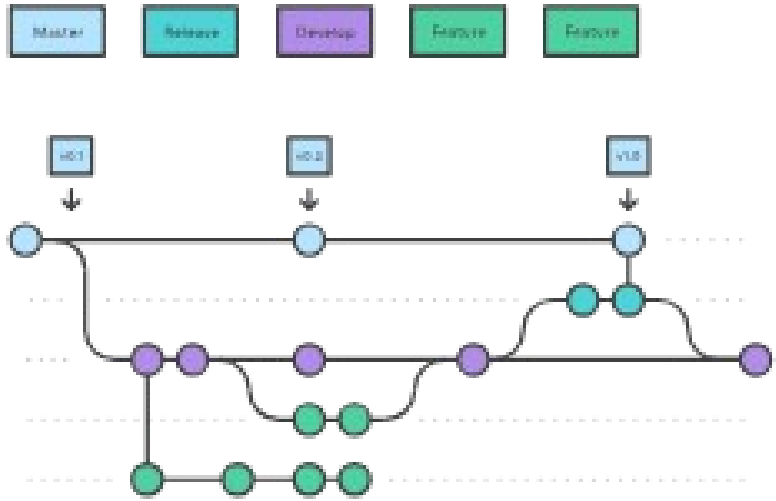
MALMÖ UNIVERSITY

Release branches

Används för att hantera releaser

Viktiga punkter:

- Avgrenas alltid från *develop*
- Slås alltid samman med *develop* och *master*
- Bör alltid heta *release/<identifier>*, t.ex. *release/1.0*



MALMÖ UNIVERSITY

Release branches

Skapa en release-branch

- Med *git-flow*

```
git flow release start identifier
```

- Standard git

```
git checkout develop
git checkout -b release/identifier
```

MALMÖ UNIVERSITY

Release branches

Avsluta en release-branch

- Med *git-flow*

```
git flow release finish identifier
```

- Standard git

```
git checkout master
git merge --no-ff release/identifier
git tag -a identifier
git checkout develop
git merge --no-ff release/identifier
git branch -d release/identifier
```

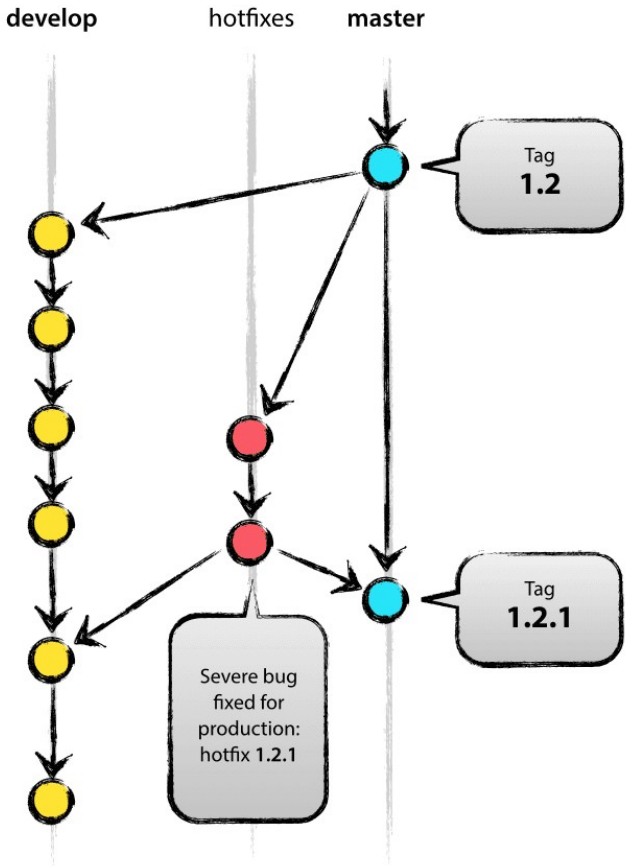
MALMÖ UNIVERSITY

Hotfix branches

Används för att fixa kritiska fel i produktionskod

Viktiga punkter:

- Avgrenas alltid från *master*
- Slås alltid samman med *master* och *develop* OM DET INTE finns en aktuell release-gren. Om det finns: slå samman med *master* och *release*
- Bör alltid heta hotfix/<*identifier*>, t.ex. hotfix*/1.0.1*



MALMÖ UNIVERSITY

Hotfix branches

Skapa en hotfix-branch

- Med *git-flow*

```
git flow hotfix start identifier
```

- Standard git

```
git checkout master
git checkout -b hotfix/identifier
```

MALMÖ UNIVERSITY

Hotfix branches

Avsluta en hotfix-branch

- Med *git-flow*

```
git flow hotfix finish identifier
```

- Standard git

```
git checkout master
git merge --no-ff hotfix/identifier
git tag -a identifier
git checkout develop
git merge --no-ff hotfix/identifier
git branch -d hotfix/identifier
```

MALMÖ UNIVERSITY

Kritik mot Gitflow

- Kan vara svårt för nya användare
- Lite för komplext för många fall (Samokhin, 2018, Ruka, 2017)
- Långlivade grenar tenderar att leda till integrationsproblem (Samokhin, 2018)
- “Buskig” historik kan vara svår att följa (Ruka, 2017)

Samokhin, Vadim. *Gitflow is a Poor Branching Model Hack*. 2018. Source: https://hackernoon.com/gitflow-is-a-poor-branching-model-hack-d46567a156e7
Ruka, Adam. *Gitflow considered harmful*. 2017. Source: https://www.endoflineblog.com/gitflow-considered-harmful

Mer om Gitflow

- En piffig latund:
  http://danielkummer.github.io/git-flow-cheatsheet/
- GitHubs genomgång av Git/Github:
  https://guides.github.com/activities/hello-world/
- Atlassians genomgång av Gitflow:
  https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

MALMÖ UNIVERSITY