

Модели ветвления в Git

Балабанов Максим

ВШЭ ФКН ПИ 2 курс

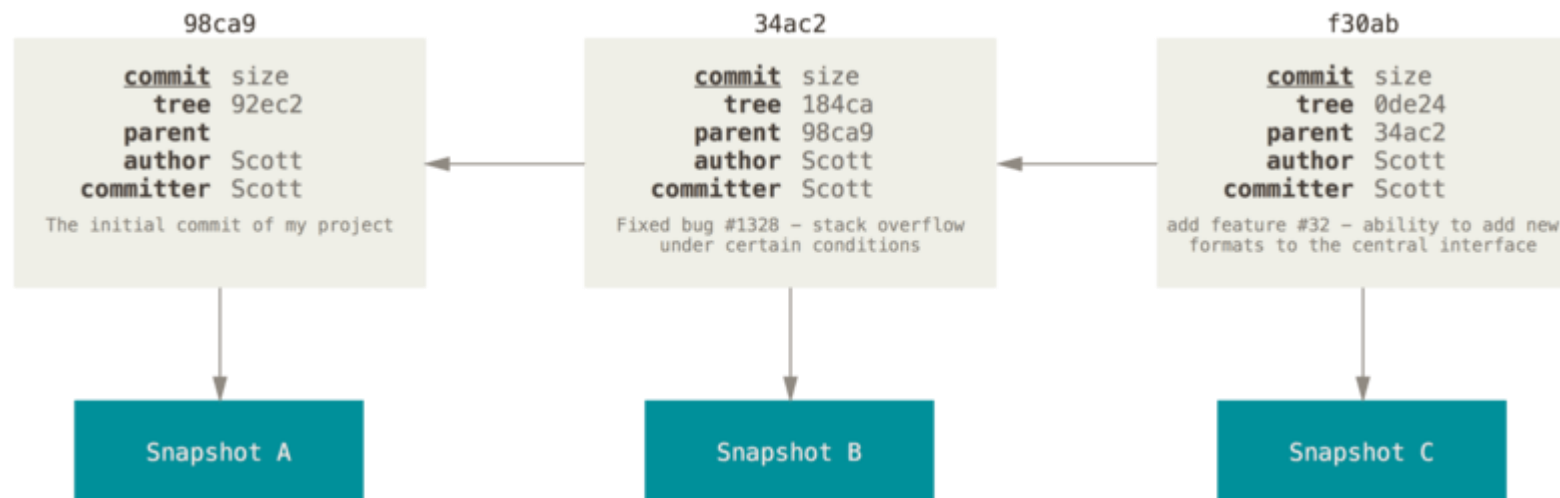
НИС

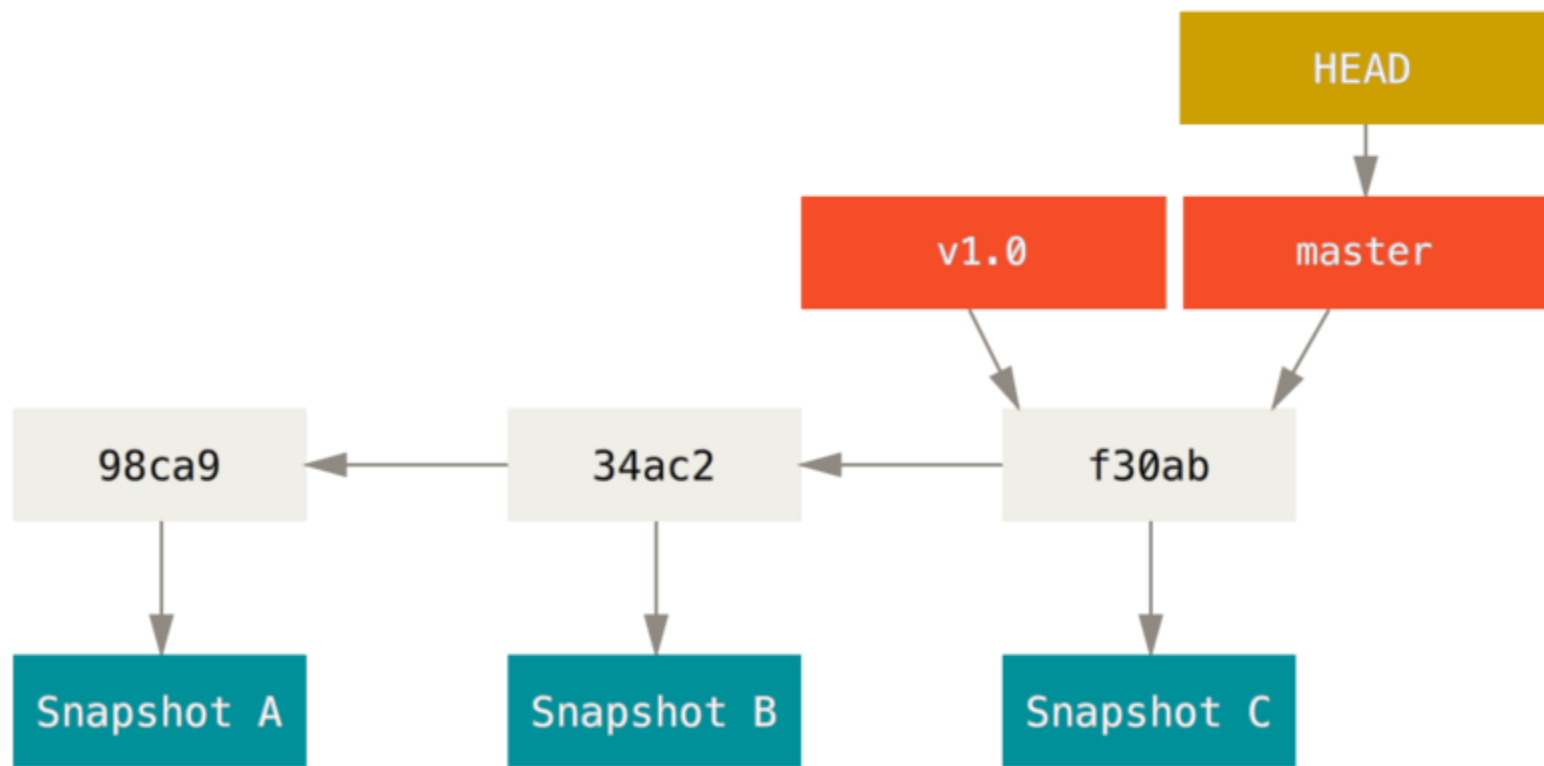
Методологии

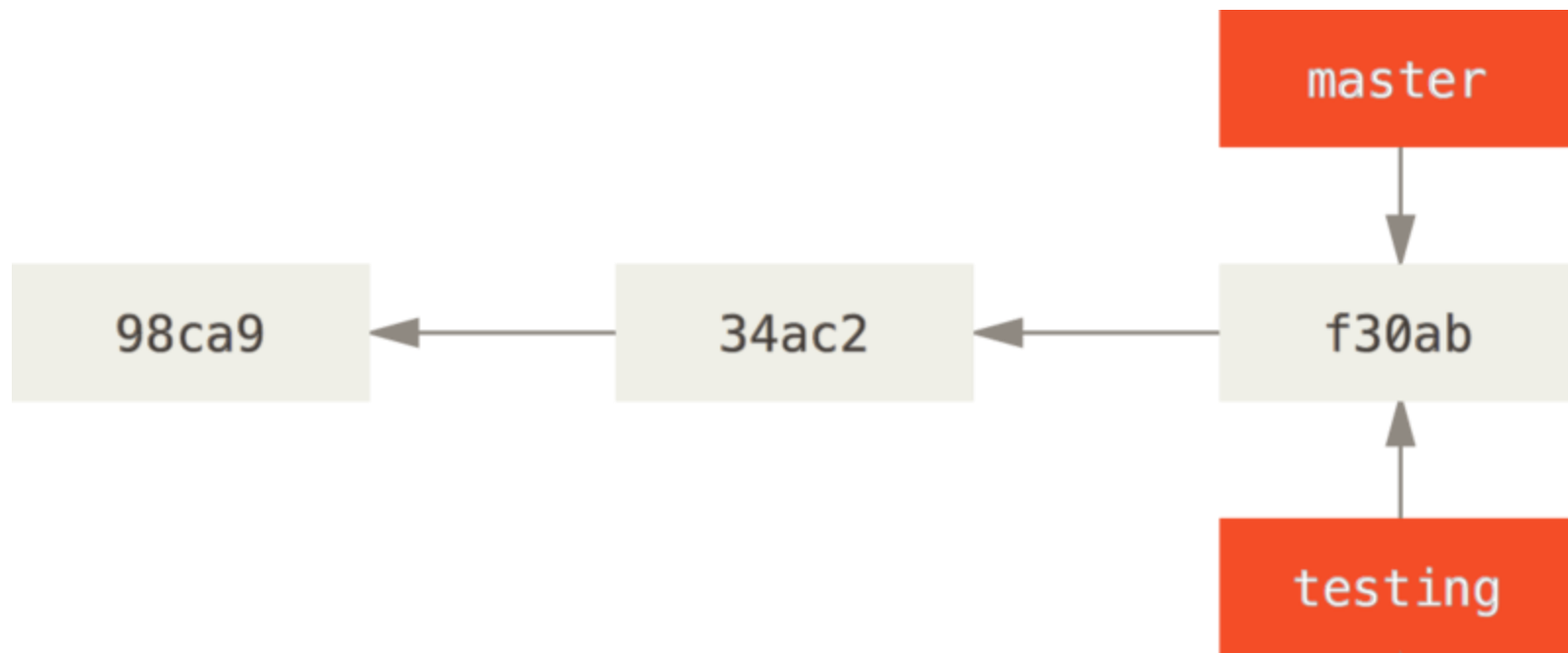
разработки ПО

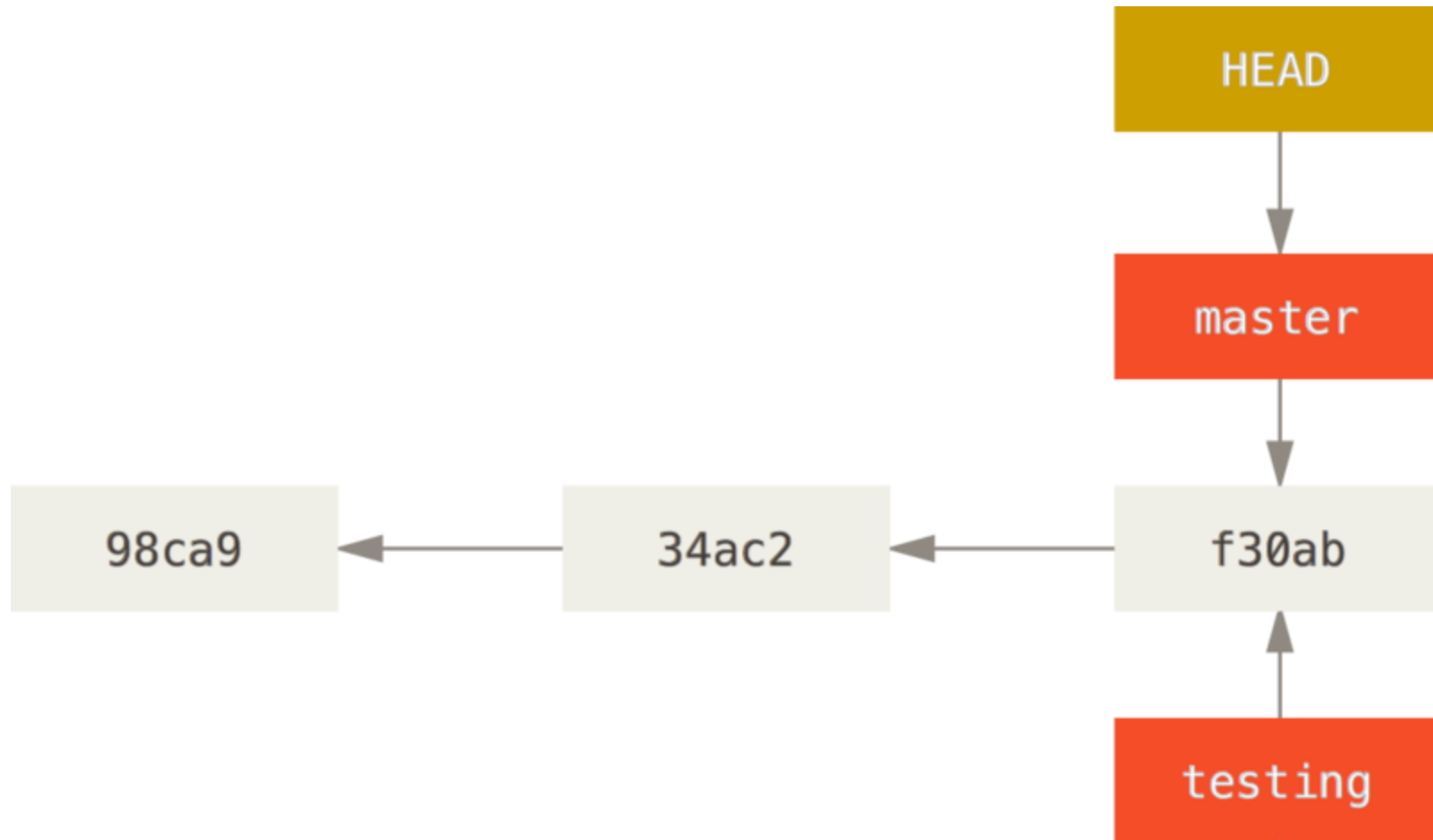


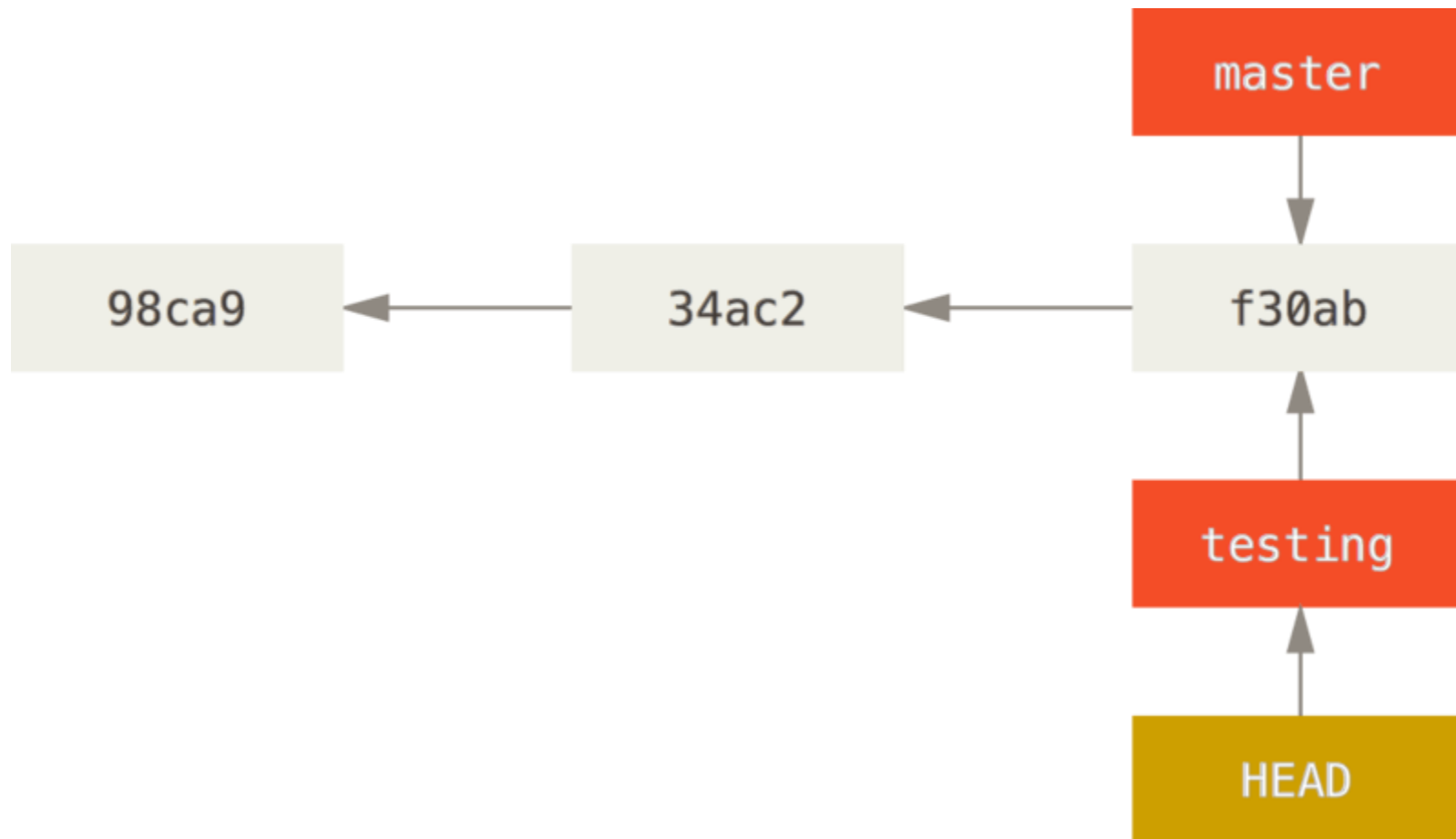


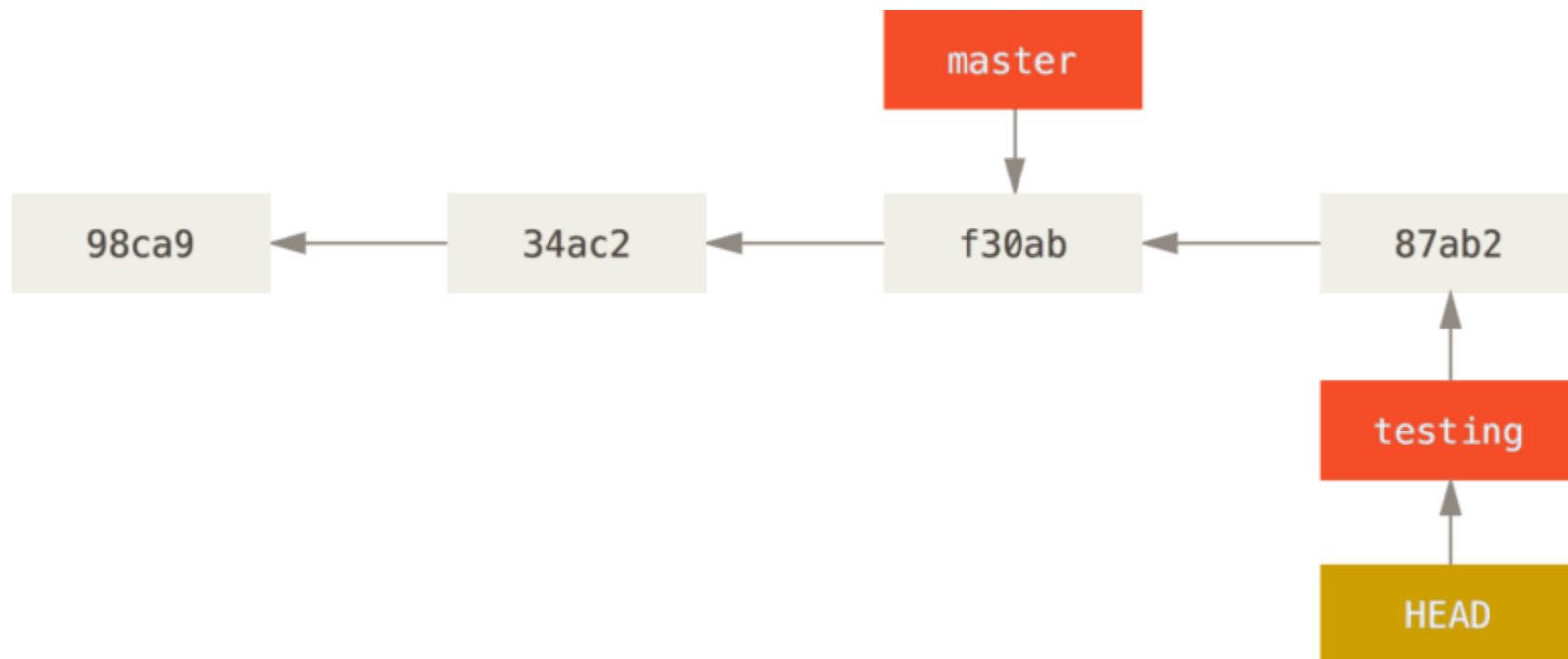


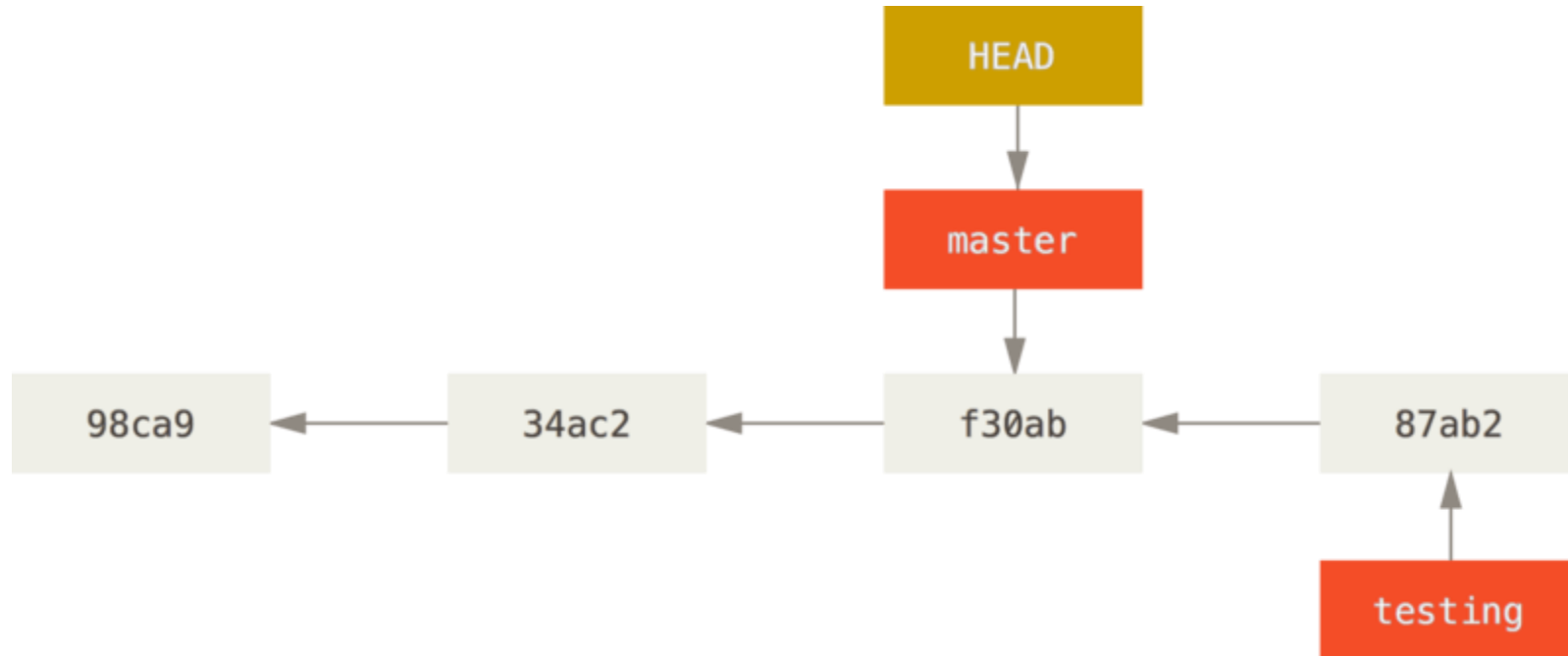


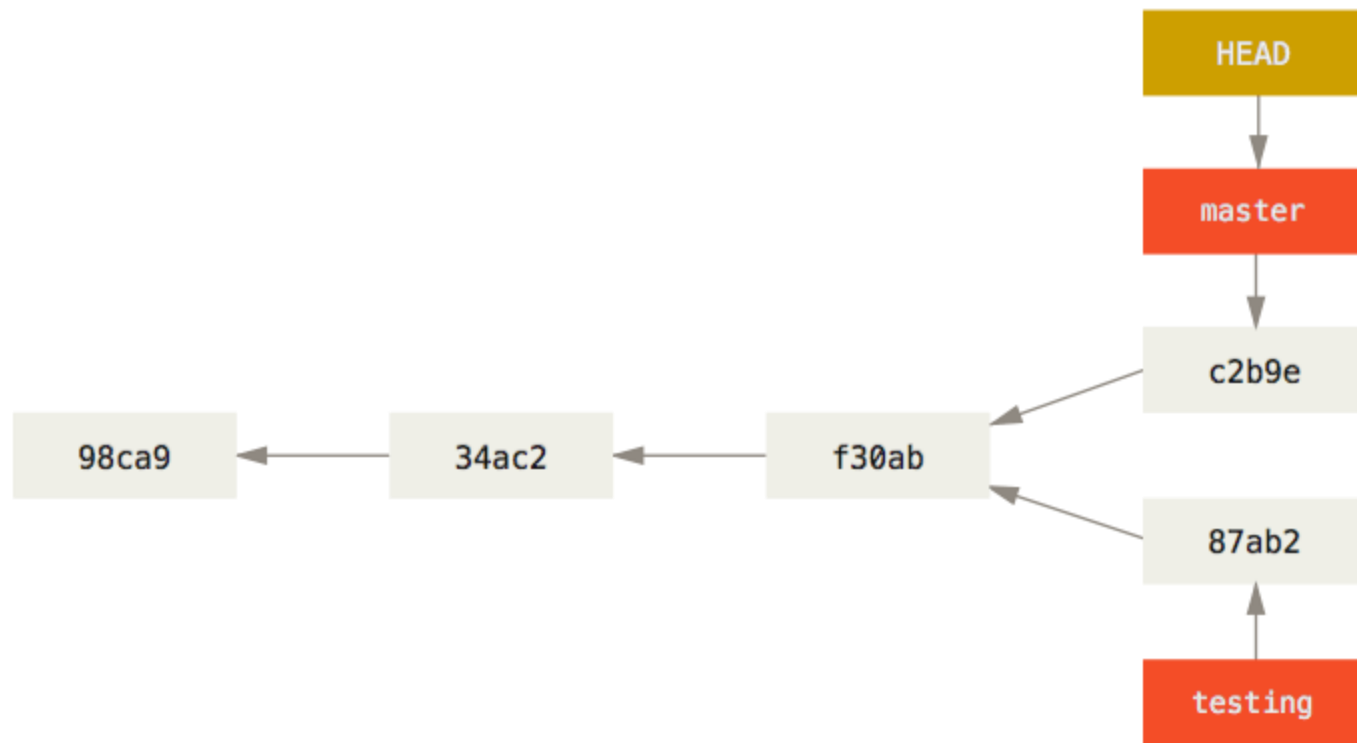




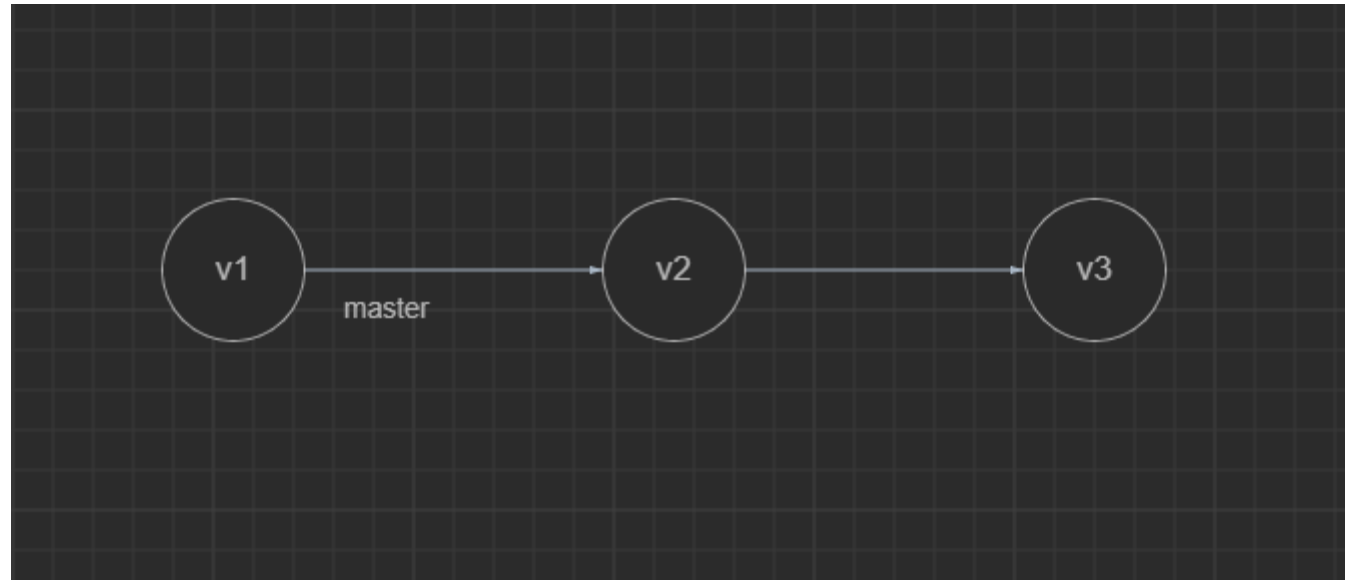




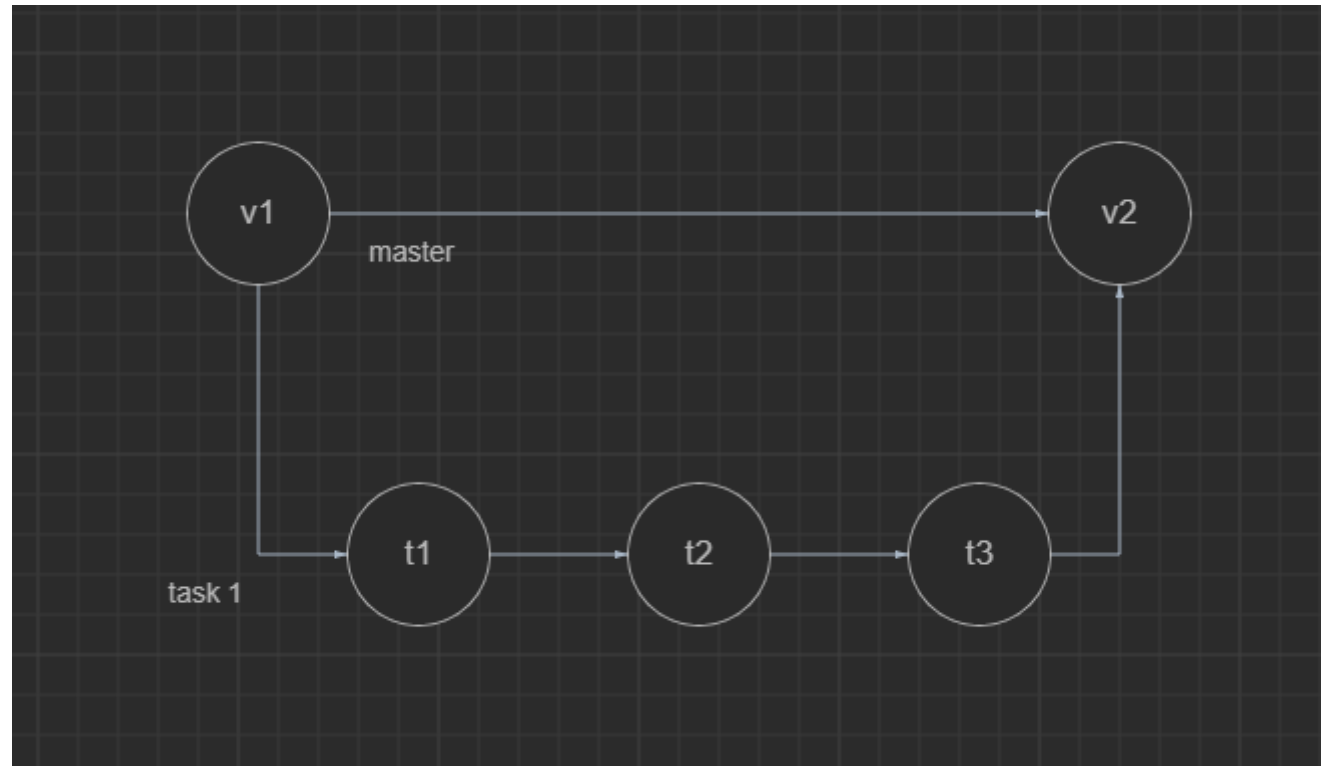




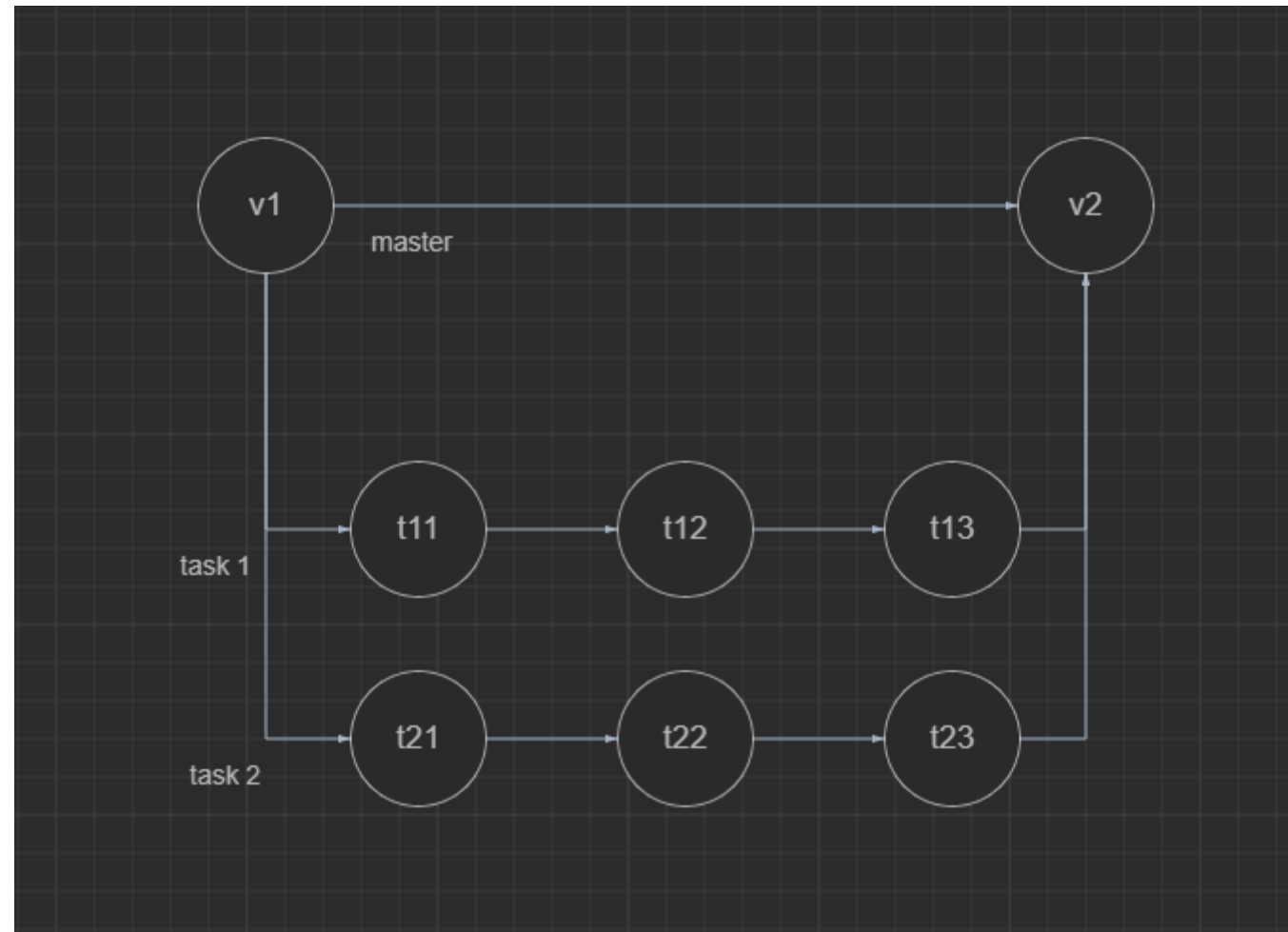
Central Model



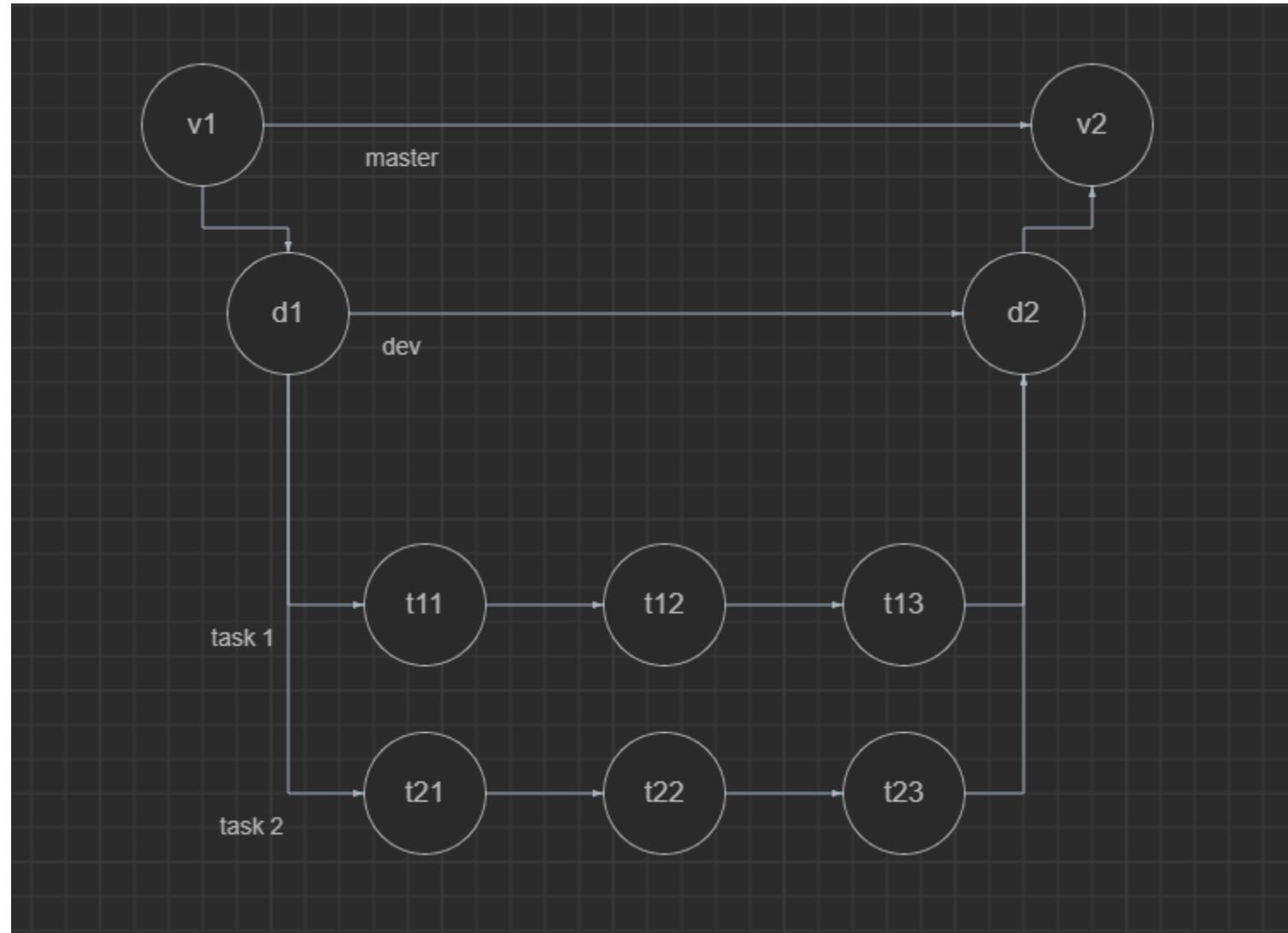
Developer Branch Model



Developer Branch Model



Feature Branch Model



A successful Git branching model



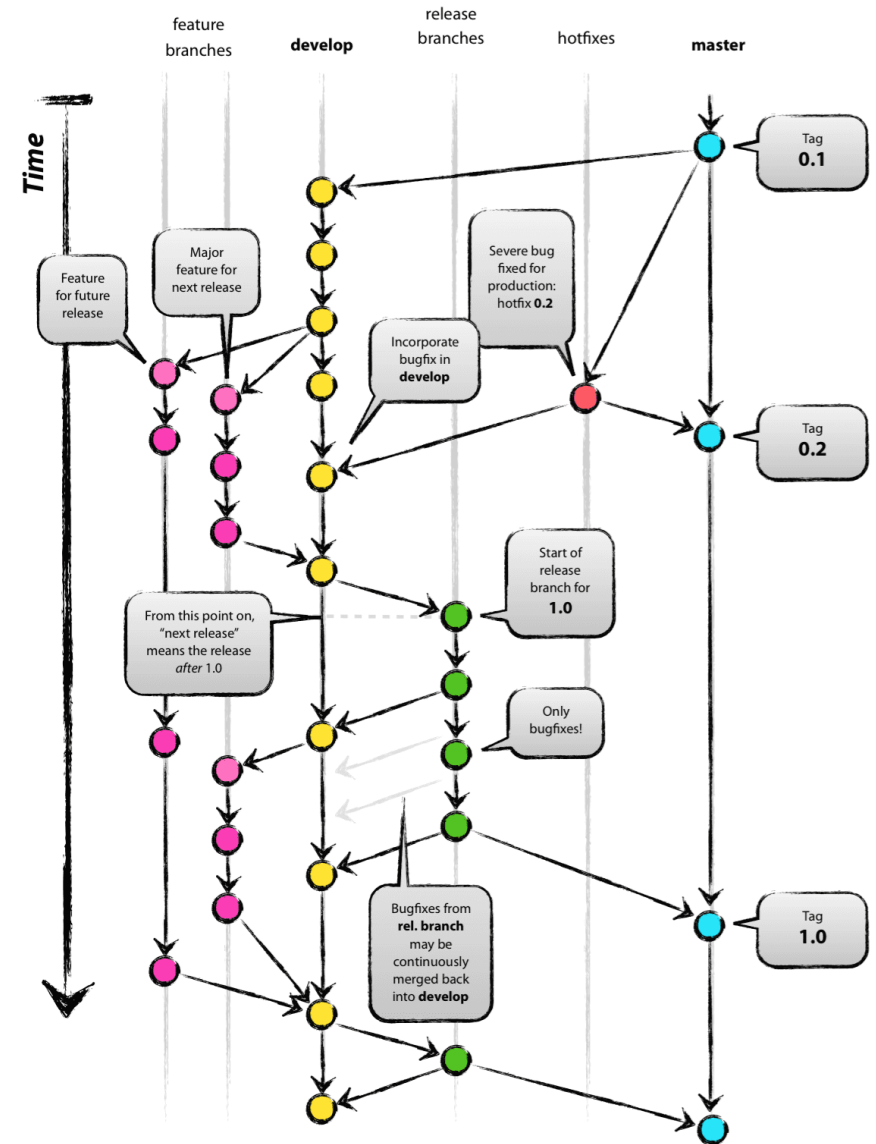
By Vincent Driessen

on Tuesday, January 05, 2010

Note of reflection (March 5, 2020)

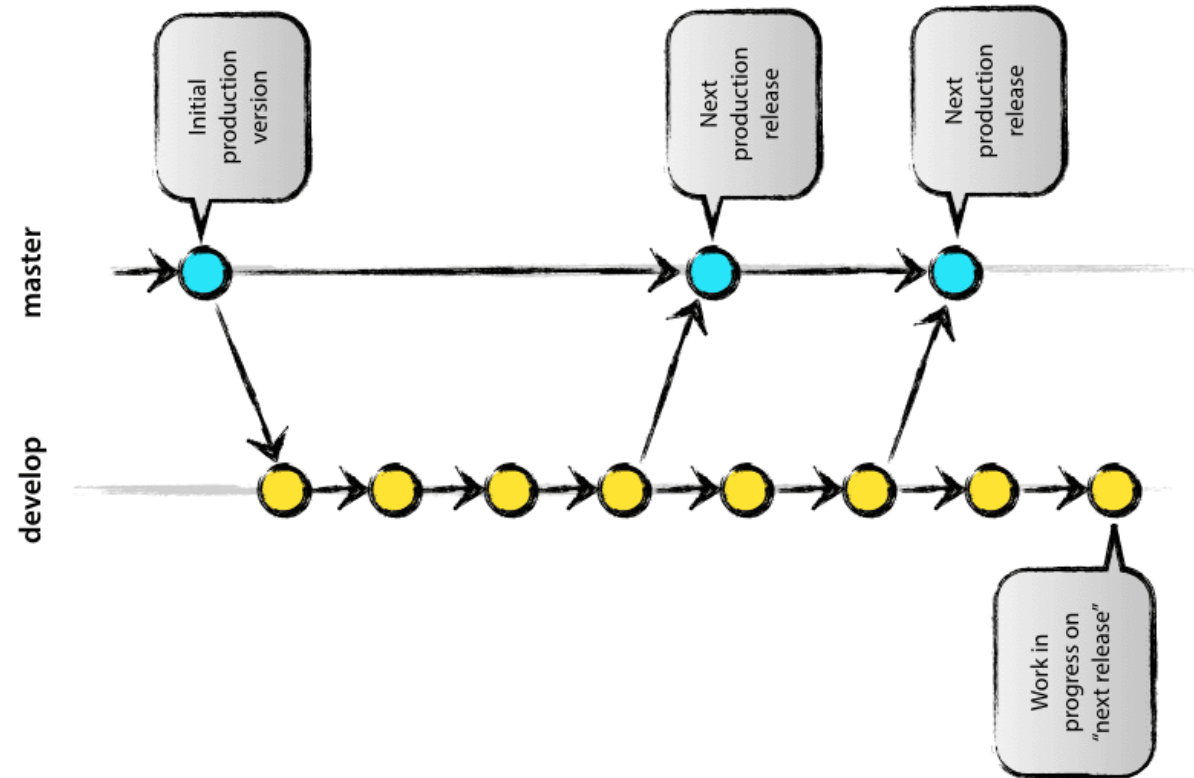
This model was conceived in 2010, now more than 10 years ago, and not very long after Git itself came into being. In those 10 years, git-flow (the branching model laid out in this article) has become hugely popular in many a software team to the point where people have started treating it like a standard of sorts — but unfortunately also as a dogma or panacea.

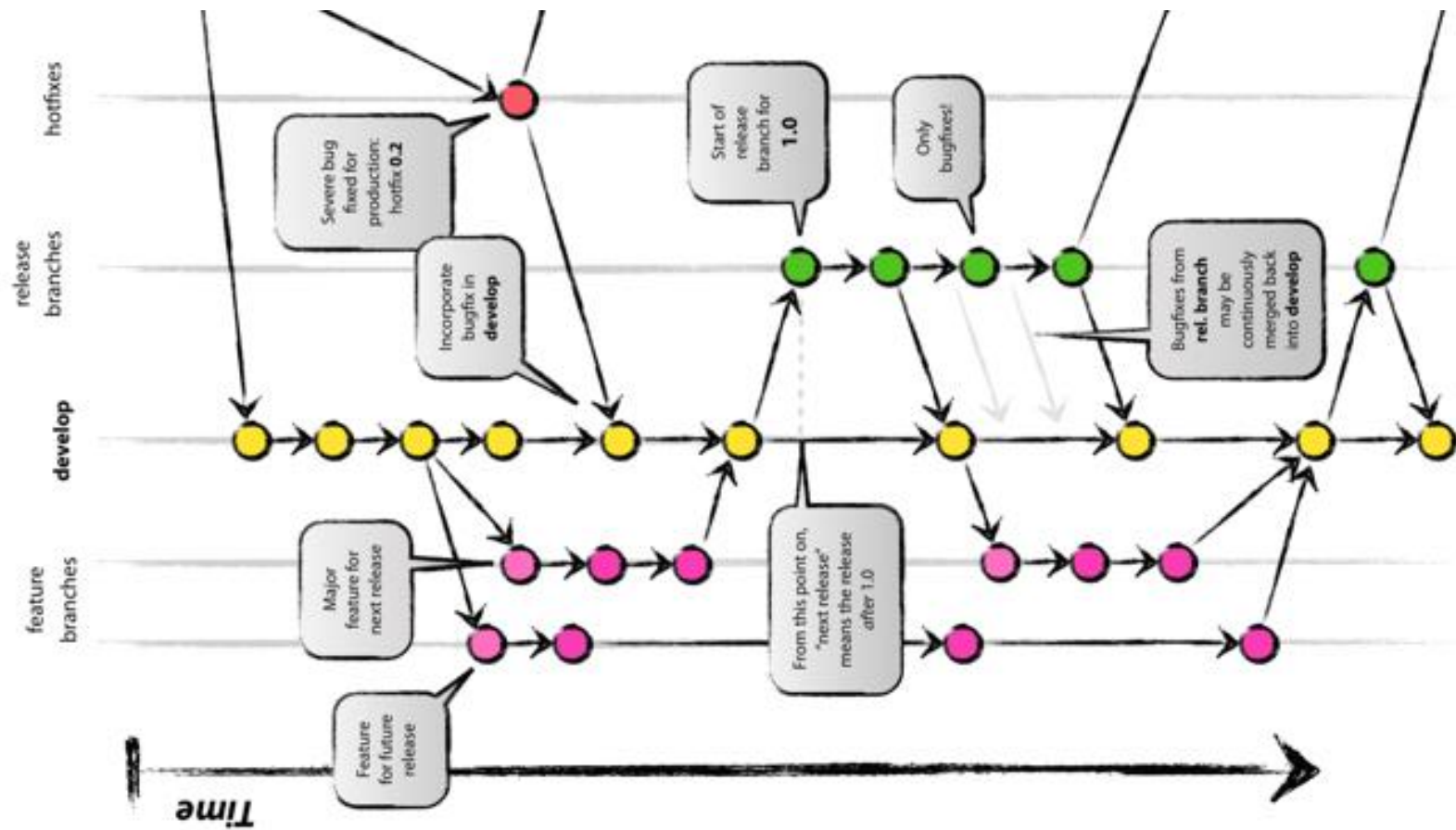
During those 10 years, Git itself has taken the world by a storm, and the most popular type of software that is being developed with Git is shifting more towards web apps — at least in my filter bubble. Web apps are typically continuously delivered, not rolled back, and you don't have to support multiple versions of the software running in the wild.



Feature Branch Model

“Successful” Branch Model





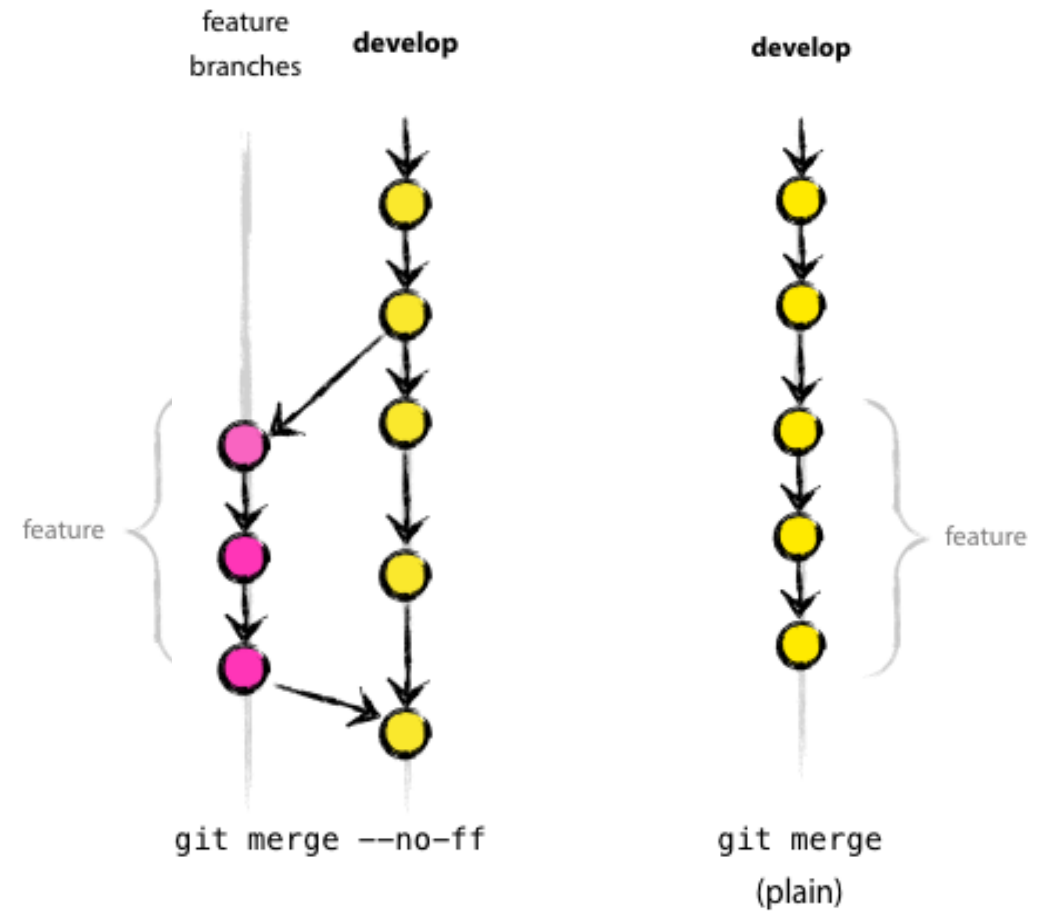
hotfixes branches

release branches

feature branches

feature branches

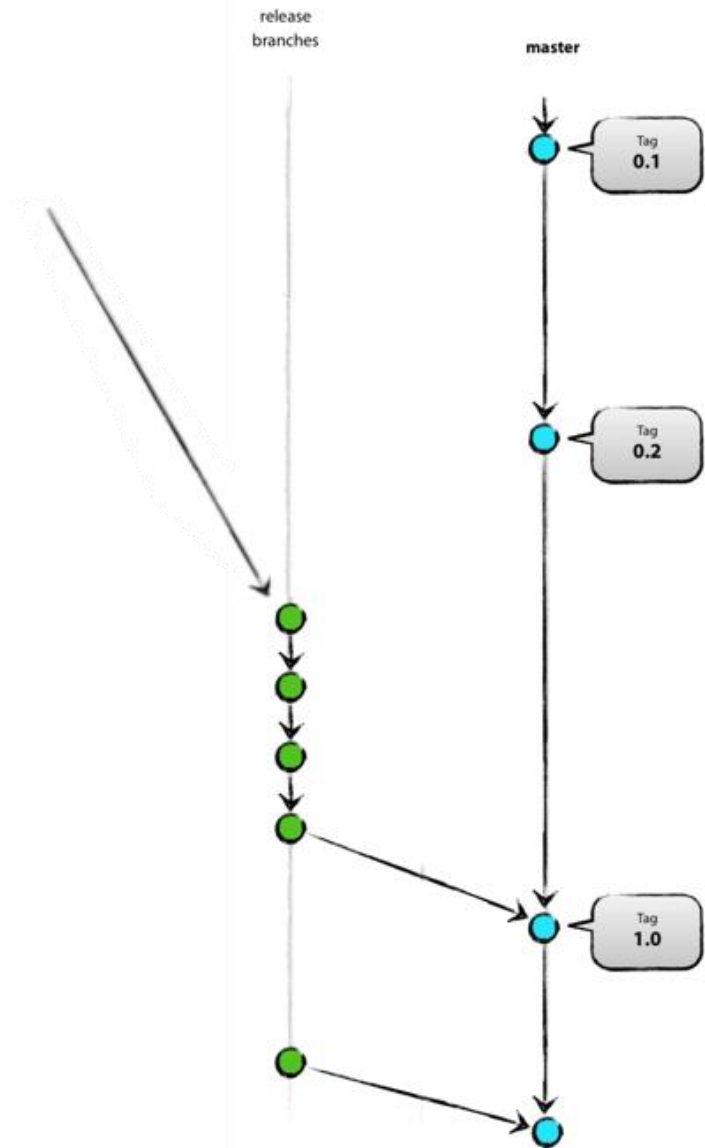
Могут порождаться от: develop
Должны вливаться в: develop



release branches

Могут порождаться от: develop

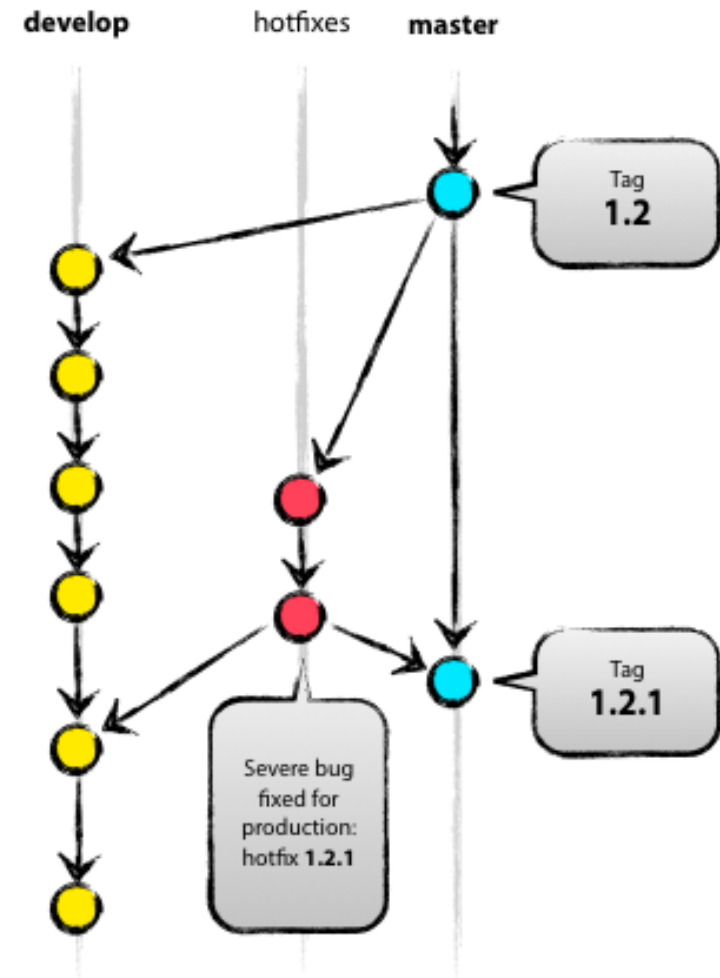
Должны вливаться в: develop и master



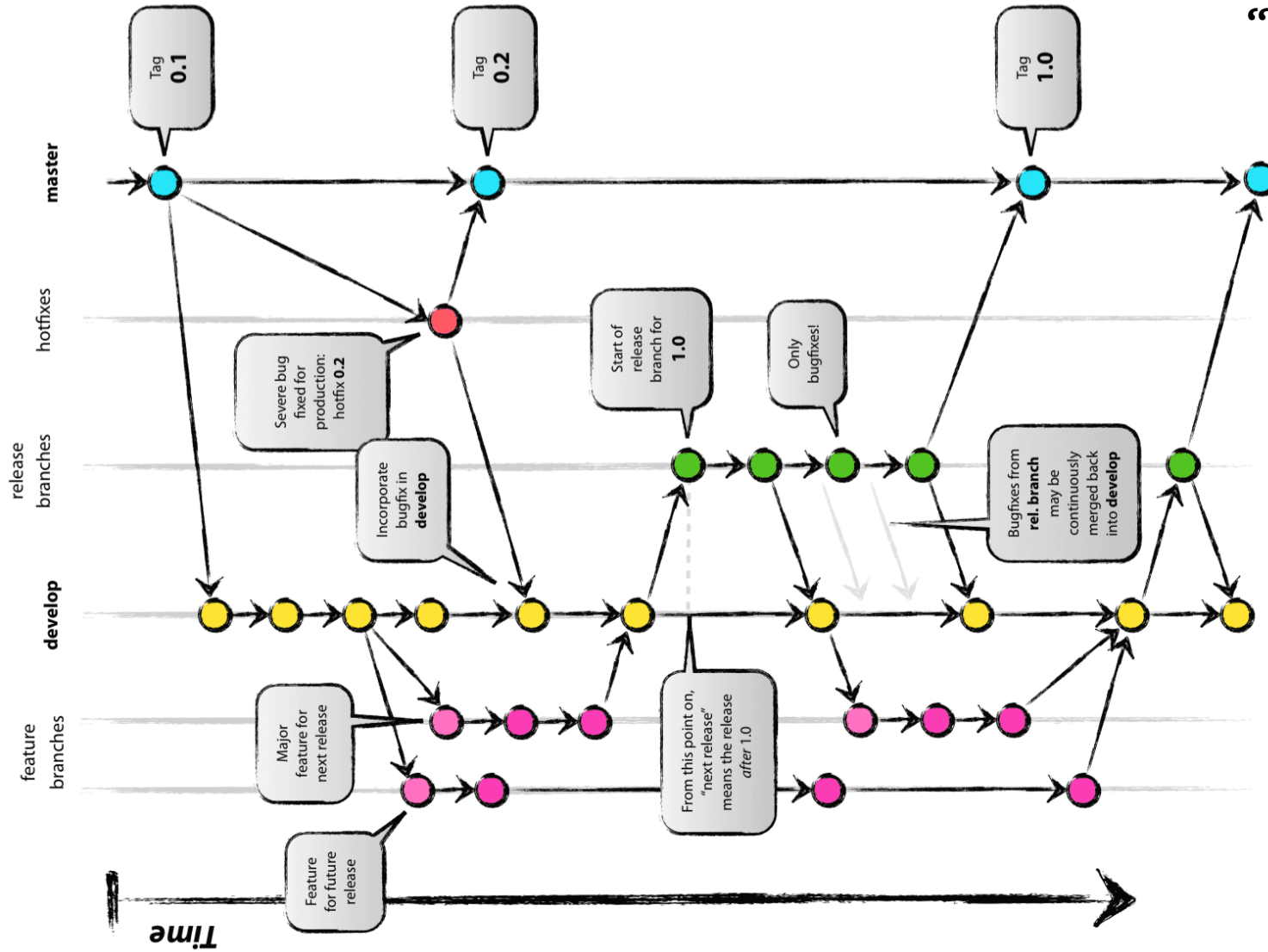
hotfix branches

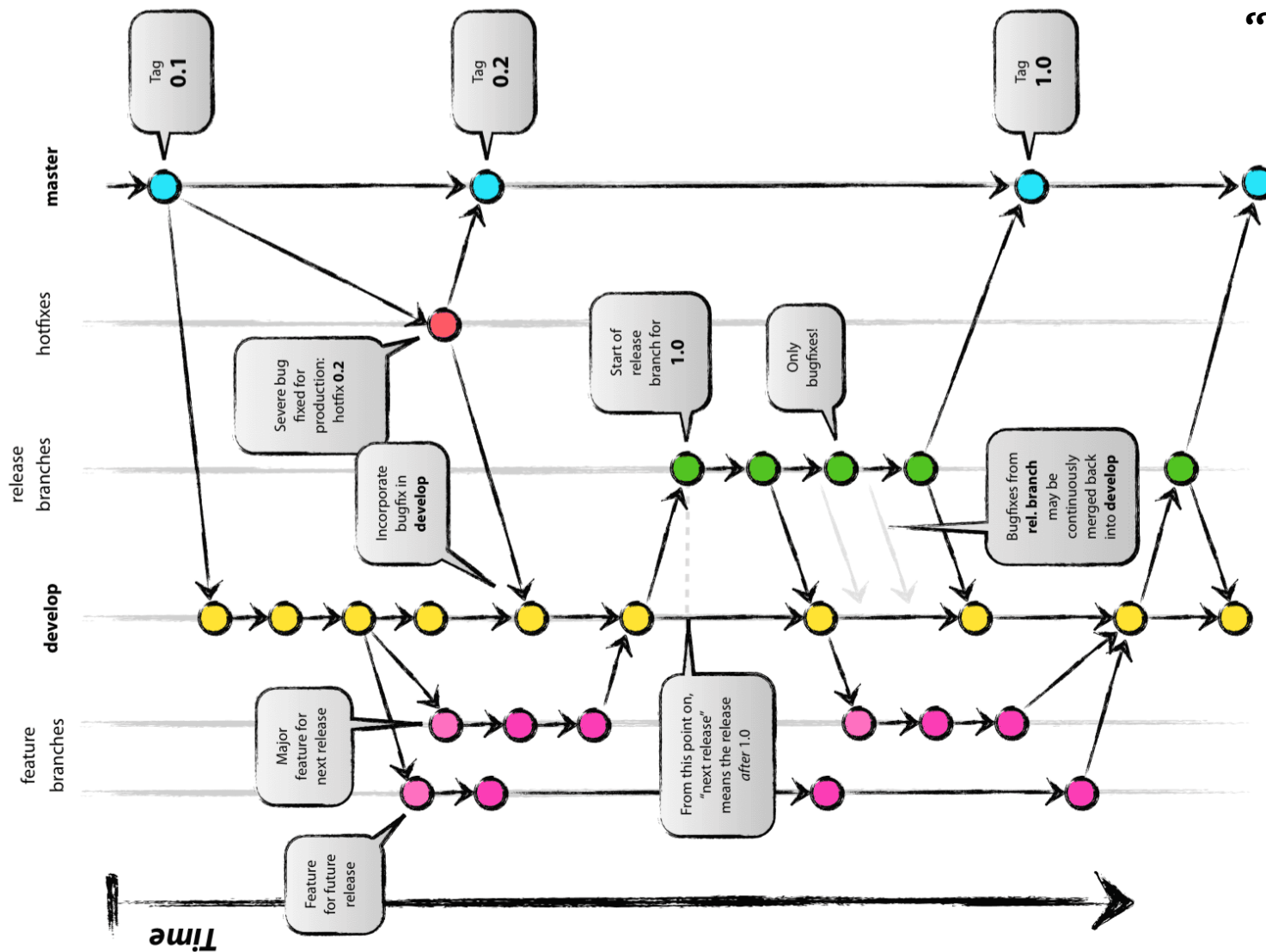
Могут порождаться от: master

Должны вливаться в: develop и master



“Successful” Branch Model





“Successful” Branch Model

Материалы:

[Оригинальная статья](#)

[Перевод статьи](#)

[Модели ветвления](#)

[Антипаттерны ветвления](#)