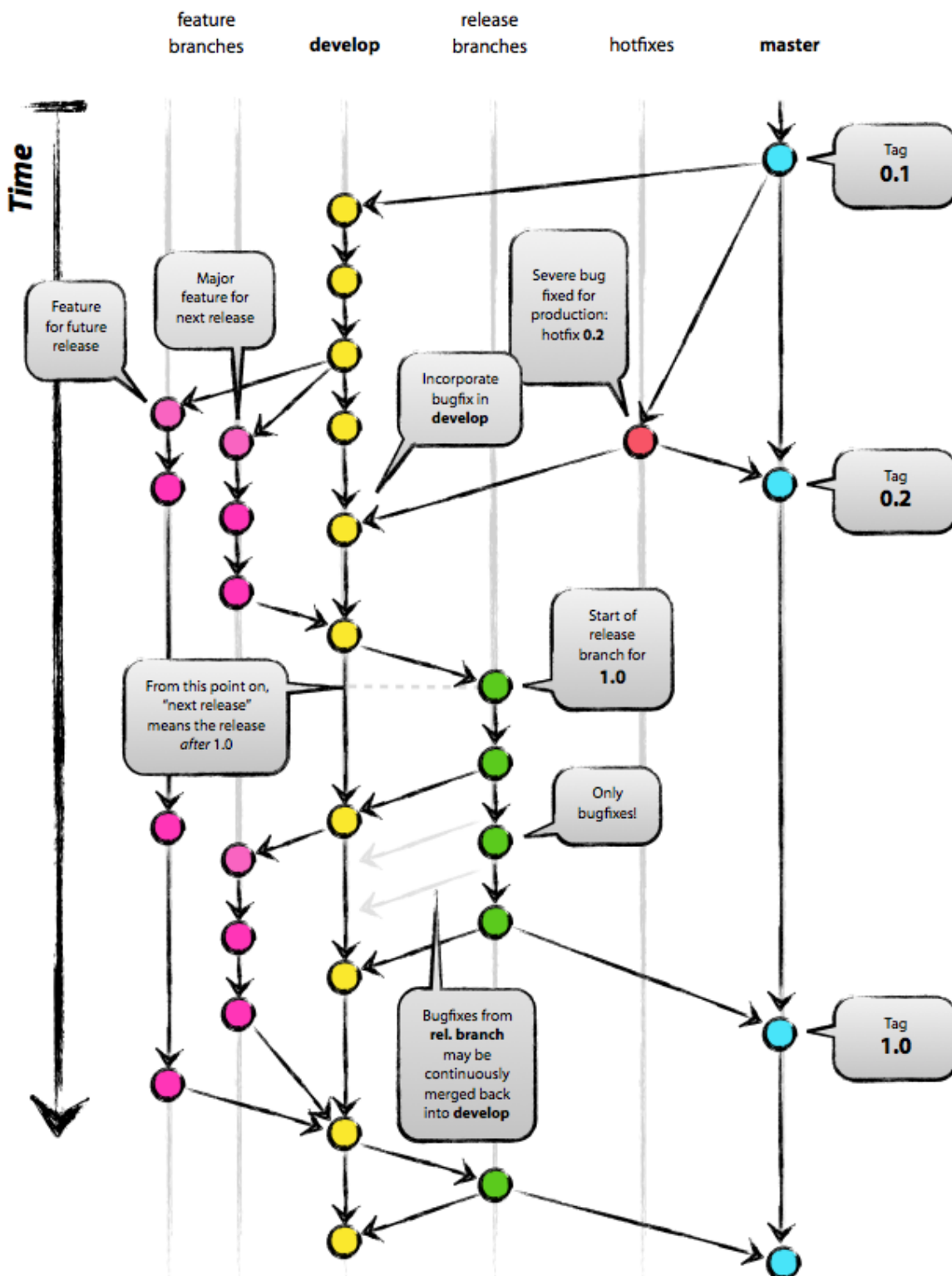


## Модель ветвления Git Flow

Для эффективной работы с git и избежания путаницы в наименовании веток часто используется общее соглашение о наименовании веток под названием «модель git flow».

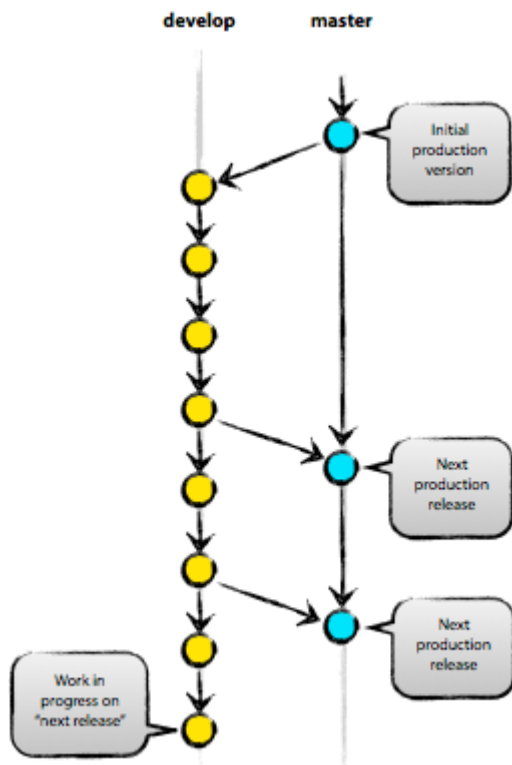


Общая схема модели git flow

## Главные ветви

Ядро модели разработки не отличается от большинства существующих моделей. Центральный репозиторий содержит две главные ветки, существующие всё время:

- master
- develop



Ветви master и develop

Ветвь master создаётся при инициализации репозитория, параллельно ей также мы создаём ветку для разработки под названием develop.

Мы считаем ветку origin/master главной. То есть, исходный код в ней должен находиться в состоянии, готовом к выпуску продукта (production-ready) в любой произвольный момент времени.

Ветвь origin/develop мы считаем главной ветвью для разработки. Хранящийся в ней код в любой момент времени должен содержать самые последние изданные изменения, необходимые для следующего релиза.

Когда исходный код в ветви разработки (develop) достигает стабильного состояния и готов к релизу, все изменения должны быть определённым способом влиты в главную ветвь (master) и помечены тегом с номером релиза.

## Вспомогательные ветви

Помимо главных ветвей master и develop, модель разработки содержит некоторое количество типов вспомогательных ветвей, которые используются для распараллеливания разработки между членами команды, для упрощения внедрения нового функционала (features), для подготовки релизов и для быстрого исправления проблем в производственной версии приложения.

В отличие от главных ветвей, эти ветви всегда имеют ограниченный срок жизни. Каждая из них в конечном итоге рано или поздно удаляется.

Используемые типы ветвей:

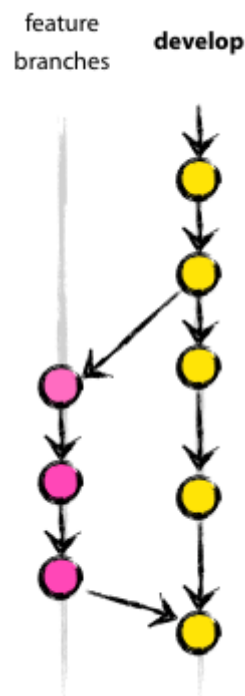
- Ветви функциональностей (Feature branches);
- Ветви релизов (Release branches);
- Ветви исправлений (Hotfix branches).

### Ветви функциональностей (feature branches)

Могут порождаться от: develop

Должны вливаться в: develop

Соглашение о наименовании: всё, за исключением master, develop, release-\* или hotfix-\*.



Ветви новой  
функциональности

*Ветви функциональностей* (feature branches), также называемые тематическими ветвями (topic branches), используются для разработки новых функций, которые должны появиться в текущем или будущем релизах. При начале работы над функциональностью (фичей) может быть ещё неизвестно, в какой именно релиз она будет добавлена. Смысл существования ветви функциональности (feature branch) состоит в том, что она живёт так долго, сколько продолжается разработка данной функциональности (фичи). Когда работа в ветви завершена, последняя вливается обратно в главную ветвь разработки (что означает, что функциональность будет добавлена в грядущий релиз) или же удаляется (в случае неудачного эксперимента).

## Ветви релизов (release branches)

Могут порождаться от: develop

Должны вливаться в: develop и master

Соглашение о наименовании: release-\*

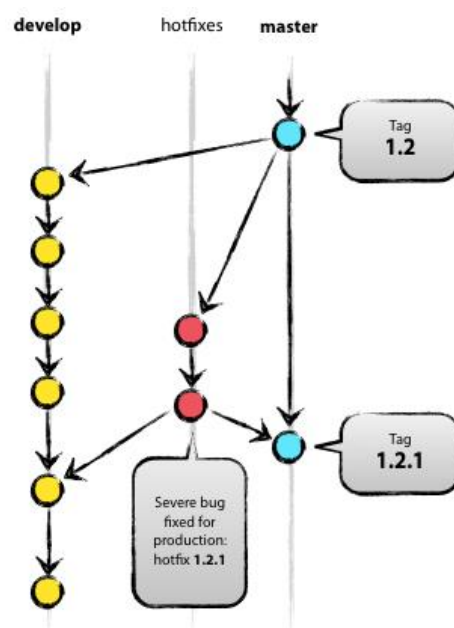
Ветви релизов используются для подготовки к выпуску новых версий продукта. Они позволяют расставить финальные точки над *i* перед выпуском новой версии. Кроме того, в них можно добавлять минорные исправления, а также подготавливать метаданные для очередного релиза (номер версии, дата сборки и т.д.).

## Ветви исправлений (hotfix branches)

Могут порождаться от: master

Должны вливаться в: develop и master

Соглашение о наименовании: hotfix-\*



Ветви исправлений

Ветви для исправлений порождаются необходимостью немедленно исправить нежелательное поведение производственной версии продукта. Когда в производственной версии находится баг, требующий немедленного исправления, из соответствующего данной версии тега главной ветви (master) порождается новая ветвь для работы над исправлением.

Вывод:

Подобная модель ветвления помогает лучше организовать проект и избавляет от несогласованности в именах ветвей. Сейчас во многих компаниях формат наименования git flow является стандартом де-факто.