

Разработка пользовательского веб-интерфейса

GIT, GitLab, GitHub

Сергей Геннадьевич Синица

КубГУ, 2023

sin@kubsu.ru

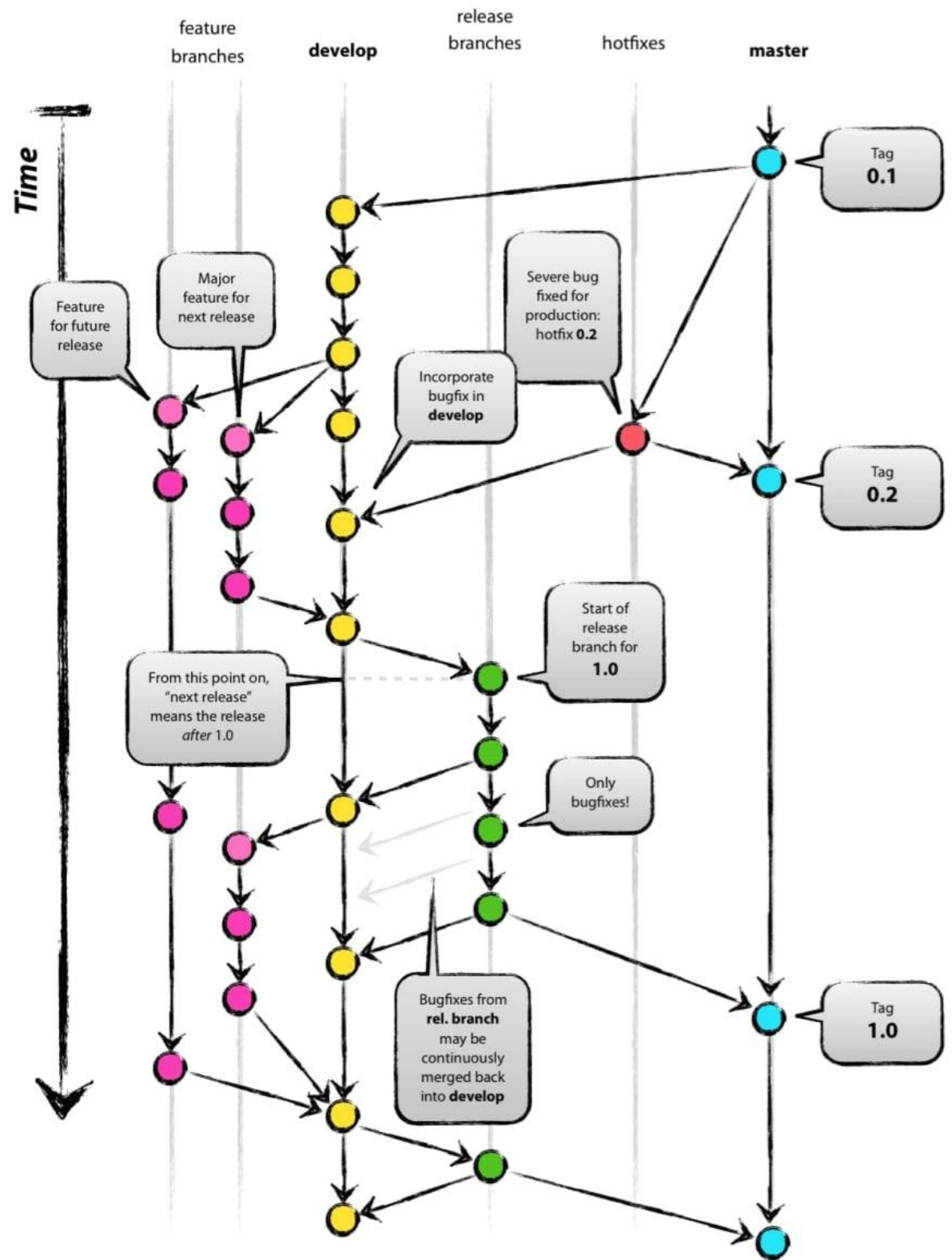
НУ ДАВАЙ, РАССКАЖИ МНЕ

КАК ТЫ КОДИШЬ БЕЗ КОНТРОЛЯ ВЕРСИЙ



Линус Торвальдс, создатель Git и Linux, передает привет Nvidia

GITFLOW



НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ

**И ЗАМЕРДЖИТЬ В МАСТЕР
БЕЗ ПУЛ РЕКВЕСТА**

В Linux и MacOS устанавливаем GIT. В Windows устанавливаем GitBash.

Сначала генерируем пару открытый/закрытый ключ и настраиваем авторизацию в GitHub или GitLab:

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>
<https://docs.gitlab.com/ee/user/ssh.html>

Клонирование репозитория с GitLab по HTTPS:

```
git clone https://gitlab.com/s.sinitsa/web1.git
```

Вместо s.sinitsa подставить логин на гитлабе, вместо web1 — название проекта.

Клонирование репозитория с GitHub по SSH:

```
git@github.com:mrsin/mrsin.github.io.git
```

Вместо mrsin подставить логин на гитхабе, вместо mrsin.github.io — название проекта.

При клонировании GitHub репы по https работать будет только на чтение! Создается каталог проекта с полной копией всех файлов и историей всех коммитов. В каталоге .git хранится сам репозиторий, остальное — файлы рабочей копии проекта. Все дальнейшие команды надо выполнять в каталоге проекта. Репозиторий «помнит» откуда он склонирован и git может «подтянуть новые коммиты» (загрузить изменения) или отправить сделанные в рабочей копии коммиты с помощью команд git pull и git push соответственно.

Команда `git status` показывает изменения в файлах по сравнению с текущим коммитом в текущей ветке. По умолчанию текущая ветка `main`. Она соответствует коду сайта на сервере.

Если в рабочей копии изменить файлы, то для фиксации изменений в `git` необходимо добавить измененные файлы в коммит:

`git add *`

Создать коммит с этими изменениями и отправить его в репозиторий GitLab:

`git commit -m "Комментарий"`

`git push`

Команда **`git log`** покажет историю коммитов. Каждый коммит имеет уникальный идентификатор (хеш).

Команда **`git revert`** хеш_комита откатит изменения коммита и создаст новый коммит.

Более полная инструкция к использованию GIT совместно с GitLab:
<https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html>

Создание новой ветки:

```
git branch имя_ветки
```

Переключение текущей ветки:

```
git checkout имя_ветки
```

Вливание в текущую ветку другой веткой:

```
git merge имя_вливаемой_ветки
```

Удаление ветки:

```
git branch -d имя_ветки
```