

Разработка пользовательского веб-интерфейса

GIT

Сергей Геннадьевич Синица

КубГУ, 2020

sin@kubsu.ru

A close-up shot of Gene Wilder as Willy Wonka. He is wearing a brown top hat, a purple velvet jacket, a white shirt, and a large tan bow tie. He has a mischievous, knowing smile and is resting his chin on his right hand. The background is slightly out of focus, showing what appears to be a stone wall and a yellow vertical pole.

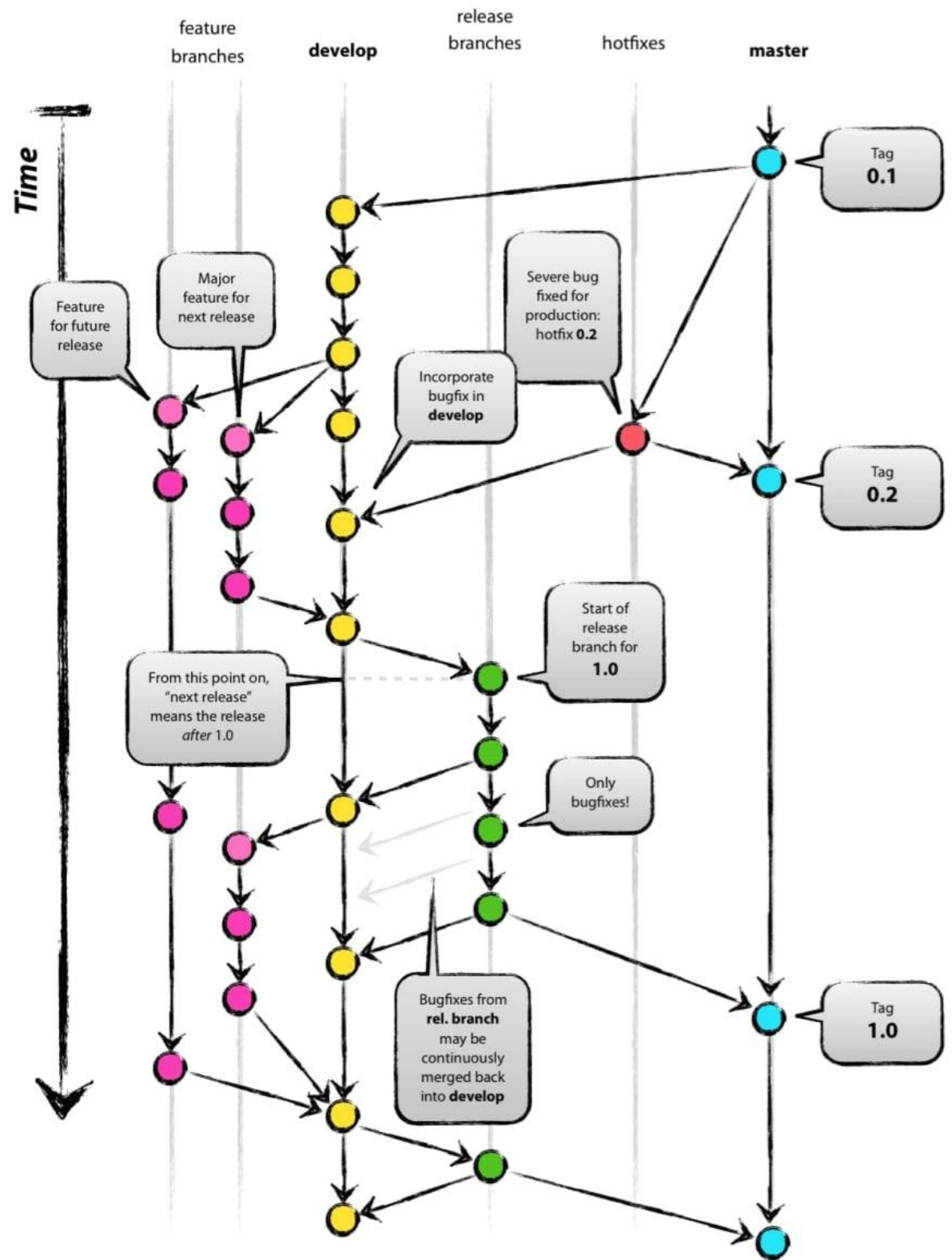
НУ ДАВАЙ, РАССКАЖИ МНЕ

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**

Клонирование репозитория:

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

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

Создается каталог web1 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 (web1).

Репозиторий «помнит» откуда он клонирован 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 master. 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
```

Слияние текущей ветки с другой веткой:

```
git merge имя_ветки
```

Удаление ветки:

```
git branch -d имя_ветки
```