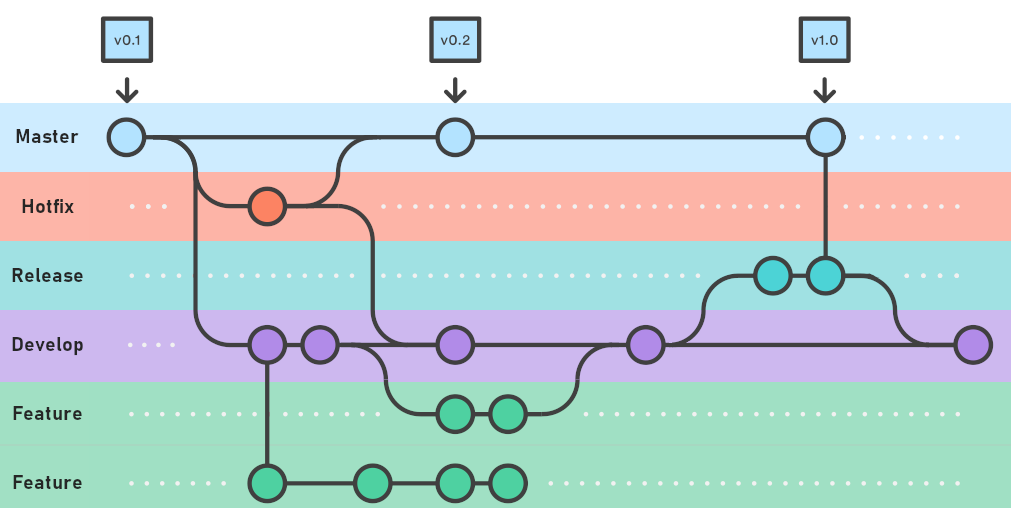
**Руководство по эксплуатации веток GIT**



Мы взяли за основу эту структуру, решили убрать hotfix и release, оставили ветку dev. Итого остались две основные ветки: master (он же main) и dev. Также под задания создаются много feature веток.

Когда мы делаем новое задание, мы создаем новую ветку под задание. И каждый обычный программист создает свой commit. Когда задание закончено, старший программист (Сергей Егоров) делает pull-request на ветку dev.

Роли:

- За merge ветки dev на main отвечает главный программист Константин Кухта.

- За принятие pull-request’ов на ветку dev с разных веток feature отвечает Сергей Егоров.

- Индивидуальные роли для выполнения заданий на ветках feature выдаются под конкретные задания обычным программистам, среди них: Екатерина Киндрук, Юлия Дубовицкая, Денис Агадашев, Геннадий Кузнецов, Кирилл Гайворонский.

Шаг 1: Создайте локальную копию репозитория

Для начала работы с проектом вам нужно создать локальную копию репозитория на своем компьютере. Для этого воспользуйтесь командой git clone, указав в качестве аргумента URL-адрес репозитория на GitHub.

bash

Copy code

git clone <URL репозитория>

Шаг 2: Создайте ветки

В Git основная ветка проекта обычно называется main или master. Создайте ветку dev для разработки функционала и дополнительные ветки feature для отдельных задач. Для создания новой ветки используйте команду git branch, указав название ветки.

bash

Copy code

git branch dev

git branch feature/task1

git branch feature/task2

Шаг 3: Переключайтесь между ветками

Для того, чтобы переключиться на другую ветку, воспользуйтесь командой git checkout, указав название ветки.

Copy code

git checkout dev

Шаг 4: Делайте изменения в коде и коммитте их

Внесите необходимые изменения в код проекта и закоммитьте их. Используйте команду git add для добавления измененных файлов в индекс, а затем git commit для коммита изменений.

Copy code

git add .

git commit -m "Добавлен новый функционал"

Шаг 5: Отправляйте изменения на GitHub

Для того, чтобы отправить изменения на GitHub, воспользуйтесь командой git push, указав название ветки.

perl

Copy code

git push origin dev

Шаг 6: Создавайте pull request

Когда вы закончили работу над задачей и готовы добавить ее в основную ветку, создайте pull request. Для этого зайдите на страницу репозитория на GitHub и выберите вкладку "Pull requests". Затем нажмите кнопку "New pull request", выберите ветки, которые вы хотите сравнить, и добавьте комментарии о внесенных изменениях.

Шаг 7: Обсуждайте изменения

Команда может обсуждать изменения в pull request. Вы можете добавлять комментарии к коду, делать замечания и запрашивать изменения. Вся эта информация будет доступна в pull request.

Шаг 8: Merge изменений

Когда изменения были проверены и одобрены, их можно слить в основную ветку. Для этого нажмите кнопку "Merge pull request" на странице pull request.

Шаг 9: Удаляйте ветки

Когда задача выполнена и изменения были слиты в основную ветку, можно удалить ветки dev и feature. Для удаления ветки используйте команду git branch -d, указав название ветки.

bash

Copy code

git branch -d dev

git branch -d feature/task1

git branch -d feature/task2

Шаг 10: Обновляйте локальную копию

Если другой член команды внес изменения в основную ветку после того, как вы создали свою локальную копию репозитория, вам нужно получить эти изменения. Для этого используйте команду git pull, указав название ветки.

css

Copy code

git pull origin main

Шаг 11: Работайте с конфликтами

Если при попытке слить изменения из основной ветки в вашу ветку возникли конфликты, вам нужно будет их разрешить вручную. Git пометит конфликтные файлы специальными маркерами, и вам нужно будет решить, какие изменения оставить и какие отклонить.

Шаг 12: Повторите процесс для новых задач

Когда вы закончили работу над одной задачей, переключитесь на другую ветку и повторите процесс для следующей задачи.

Вот некоторые из наиболее часто используемых команд Git, которые вы можете использовать для работы с ветками:

git branch - список существующих веток

git branch <branch\_name> - создание новой ветки

git checkout <branch\_name> - переключение на другую ветку

git add <file\_name> - добавление файла в индекс

git add . - добавление всех измененных файлов в индекс

git commit -m "Commit message" - коммит изменений с сообщением

git push origin <branch\_name> - отправка изменений на сервер

git pull origin <branch\_name> - получение изменений с сервера

git merge <branch\_name> - слитие изменений из другой ветки в текущую

git branch -d <branch\_name> - удаление ветки