

GitFlow - модель ветвления Git.

Ключевые идеи:

Данная модель отлично подходит для организации рабочего процесса на основе релизов.

Gitflow предлагает создание отдельной ветки для исправлений ошибок в продуктовой среде.

Последовательность работы при использовании модели Gitflow:

- 1 Из master создается ветка develop.
- 2 Из develop создаются ветки feature.
- 3 Когда разработка новой функциональности завершена, она объединяется с веткой develop.
- 4 Из develop создается ветка release.
- 5 Когда ветка релиза готова, она объединяется с develop и master.
- 6 Если в master обнаружена проблема, из нее создается ветка hotfix.
- 7 Как только исправление на ветке hotfix завершено, она объединяется с develop и master.

git rebase сжимает все изменения в один патч. Затем интегрирует патч в целевую ветку. В отличие от merge, rebase перезаписывает историю, потому что она передаётся завершённую работу из одной ветки в другую. В процессе устраняется нежелательная история.

git diff используется для вычисления разницы между любыми двумя Git деревьями. Это может быть разница между вашей рабочей копией и индексом, разница между индексом и последним коммитом (`git diff --staged`), или между любыми двумя коммитами (`git diff master branchB`).

git reset используется в основном для отмены изменений. Она изменяет указатель HEAD и, опционально, состояние индекса. Также эта команда может изменить файлы в рабочем каталоге при использовании параметра `--hard`, что может привести к потере наработок при неправильном использовании.

git fetch связывается с удалённым репозиторием и забирает из него все изменения, которых у вас пока нет и сохраняет их локально.

git revert Откатить проект до заданного коммита можно с помощью параметра `revert` и идентификатора коммита. Создастся новый коммит — копия коммита с предоставленным идентификатором — и добавится к истории текущей ветки.

git cherry-pick берёт изменения, вносимые одним коммитом, и пытается повторно применить их в виде нового коммита в текущей ветке. Эта возможность полезна в ситуации, когда нужно забрать парочку коммитов из другой ветки, а не сливать ветку целиком со всеми внесенными в нее изменениями.

git tag используется для задания постоянной метки на какой-либо момент в истории проекта. Обычно она используется для релизов.

git merge выполняет слияние коммитов из одной ветки в другую. В этом процессе изменяется только целевая ветка. История исходных веток остается неизменной.