Git Flow — это популярная модель ветвления для работы с Git, которая помогает организовать процесс командной разработки и выпуска программного обеспечения. Она структурирует работу с репозиторием, чтобы команда могла одновременно разрабатывать новые фичи, фиксить баги и готовить новый релиз без конфликтов.

**Основные ветки в Git Flow**

1. main (или master)

— Главная ветка, где лежит только стабильный, протестированный и готовый к деплою код.

2. develop

— Ветка для интеграции новых фич. То, что попадет из нее в main, обычно становится следующим релизом.

3. feature

— Ветки для разработки отдельных новых функций (feature).

4. release

— Ветки для подготовки релизов (release).

5. hotfix

— Ветки для срочного исправления багов на production (hotfix).

**Как работает Git Flow (пошагово)**

**1. Начало работы с git flow**

Обычно используется специальное расширение git-flow, но можно работать вручную.

Для инициализации git flow в проекте:

git flow init

Тебя попросят выбрать имена веток и префиксы (чаще всего можно соглашаться с дефолтными).

**2. Добавление новой фичи**

1. Отвечвляемся от develop:

git checkout develop

git checkout -b feature/my-awesome-feature

2. Пишем код, фиксируем:

git add .

git commit -m "описание изменений"

3. Когда фича готова — вливаем её обратно в `develop`:

git checkout develop

git merge feature/my-awesome-feature

или, если используешь git-flow расширение:

git flow feature finish my-awesome-feature

3. Подготовка к релизу

1. Когда нужно выкатить новый релиз, от `develop` создаётся ветка:

git checkout develop

git checkout -b release/1.0.0

2. В ней делают финальные правки: метаданные, багфиксы, подготовка к релизу.

3. Как релиз готов:

git checkout main

git merge release/1.0.0 вливание релиза в main

git tag -a 1.0.0 -m "Релиз 1.0.0" обязательно тегировать релизы!

git checkout develop

git merge release/1.0.0 # вливаем финальные правки и сюда

4. Удаляем релизную ветку:

git branch -d release/1.0.0

**4. Срочные исправления (hotfix)**

Если баг найден на production:

1. От main создается hotfix:

git checkout main

git checkout -b hotfix/1.0.1

2. Фиксим баг, коммитим:

git add .

git commit -m "Fixed critical bug"

3. Вливаем исправления в main:

git checkout main

git merge hotfix/1.0.1

git tag -a 1.0.1 -m "Hotfix 1.0.1"

4. И обязательно — в develop (чтобы фиксы не потерялись):

git checkout develop

git merge hotfix/1.0.1

5. Удаляем ветку hotfix.

- Новые фичи разрабатываются в feature/\* ветках, объединяются в develop, потом создается release.

- release вмерживается в main и back-мержится в develop.

- Если критичный баг — hotfix применяется и на main, и на develop.

**Основные Плюсы Git Flow:**

- Четкая структура, нет конфликтов между разработкой и релизами.

- Удобно для команд с длинными релизными циклами.

- Каждый знает, куда что вливать.

**Минусы:**

- Сложновато для совсем небольших или очень быстрых команд.

- Много ветвления (но это делается ради упорядоченности).

**Итоги и советы**

- Никогда не работай напрямую с main!

- Всегда тегируй релизы.

- Для экспериментов всегда создавай feature-ветки.

- После релиза не забывай back-мержить изменения обратно в develop.