

7. Стратегии ветвления

Раздел познакомит с различными стратегиями ветвления и сферами их применения.

7.1. Введение в стратегии ветвления

Выбор стратегии ветвления для репозитория обуславливается следующими факторами:

- Линейность истории и дублируемость коммитов
- Разделение кода в разработке и развертываемого кода
- Удобство построения процессов непрерывной доставки

В зависимости от предпочтений и требований команд разработки или эксплуатации может быть выбрана та или иная стратегия ветвления. В данном разделе приведены три наиболее распространенных. Любая из них является ориентиром и может быть модифицирована.

7.2. Git flow

Git flow является одной из первых документированных стратегий ветвления. Она подразумевает разделение веток на 5 основных типов:

- `main` — готовый к поставке код
- `develop` — актуальное состояния процесса разработки
- `feature` — код отдельно разрабатываемого функционала
- `release` — слепок кода для каждой поставленной версии
- `hotfix` — временное хранилище оперативных исправлений

Ветки с новым функционалом попадают в `develop`, где происходит интеграционное тестирование. В определенный момент на основе актуального состояния ветки `develop` создается ветка `release`, содержащая готовый к поставке код, который после внесения необходимых корректировок попадает в ветку `main`. Оперативные исправления основываются на ветке `main` и по готовности применяются к ней самой и к ветке `develop`.

7.3. Github flow

Github flow упрощает предыдущую схему до двух типов веток:

- `main` — готовый к поставке код
- `feature` — отдельно разрабатываемый набор изменений

Основная идея этой стратегии заключается в том, что любой новый код основывается на наиболее актуальной версии существующего, содержимое ветки `main` всегда готово к поставке, а содержимое веток `feature` постоянно тестируется, переносится в `master` по готовности и сразу развертывается. Подобный механизм удобен для небольших команд и проектов с низкой вероятностью возникновения проблем на этапе интеграции нового кода и отсутствием необходимости поддерживать несколько версий кода одновременно.

[Scott Chacon — GitHub Flow: The best way to use Git and GitHub ↗](#)

[Github Guides — Understanding the GitHub flow ↗](#)

7.4. Gitlab flow

Основное отличие Gitlab flow от Github flow заключается в наличии отдельных веток для различных окружений и релизов. Эта особенность позволяет упростить процесс доставки на несколько окружений и поддержки различных версий кода в рамках цикла поставок.

[GitLab Docs — Introduction to GitLab Flow ↗](#)

7.5. Итоги раздела

- Стратегии ветвления предназначены для формализации принципов работы
- Выбор стратегии обуславливается спецификой команды, проекта, инструментов
- Стратегии могут меняться и подстраиваться под различные требования