

# Важные дополнения

При выборе стратегии ветвления следует учитывать следующие важные факторы.

## Модульность

Модульность - это принцип построения систем, согласно которому функционально связанные части группируются в законченные узлы — модули (блоки). Модульность позволяет изменять возможности системы, путём использования/наращивания функциональных блоков, выполняющих различные задачи.

Хорошая модульность является альтернативой ветвлению, по меньшей мере, намного упрощает ветвление.

Чтобы достичь модульности, нужно постоянно следить за развитием системы и стремиться к её более модульному построению. Ключом к достижению этого является рефакторинг.

## Стабильность

Многие пользователи неохотно переходят на обновление, поскольку имеют отрицательный опыт, когда обновления закончились неудачей. Однако таким клиентам по-прежнему требуются исправления ошибок и добавление функциональности. В такой ситуации команда разработчиков сохраняет открытыми ветви продукта, который все ещё используется, и применяет к ним исправления по мере необходимости. По мере разработки становится все труднее применять исправления к старым ветвям, и это становится неприятными издержками ведения бизнеса. Снизить затраты можно только поощряя клиентов к частому обновлению до последней версии. Для этого важно поддерживать стабильность продукта.

## Конвейер развёртывания

Конвейер развёртывания - это процесс и технологическая инфраструктура, в которой изменения кода приложения переносятся из среды разработки в производственную среду. Процессы сборки, тестирования и развёртывания разбиваются на этапы и автоматизируются.

## Рефакторинг

Это процесс изменения кода, призванный упростить его обслуживание, понимание и расширение, при этом не изменяя его назначение и поведение. Термин используется по отношению к системе в целом.

## Проверка кода

Проверка кода применяется для повышения качества кода, повышения модульности, удобочитаемости и устранения дефектов в коде отдельной функции, задачи, блока. Обычно применяется перед интеграцией с магистралью.

Если команда состоит из программистов разной квалификации, то слабые разработчики показывают свою работу куратору (лидеру), который проверяет их код. Может быть организована разными способами. Например, интеграция с магистралью помечается специальным состоянием коммитов. В другом варианте, разработчики сначала интегрируются в ветку куратора.

---

Версия #5

[Демонов Сергей](#) создал Mon, Feb 5, 2024 4:13 PM

[Шаламов Сергей](#) обновил Thu, Feb 8, 2024 5:24 PM