

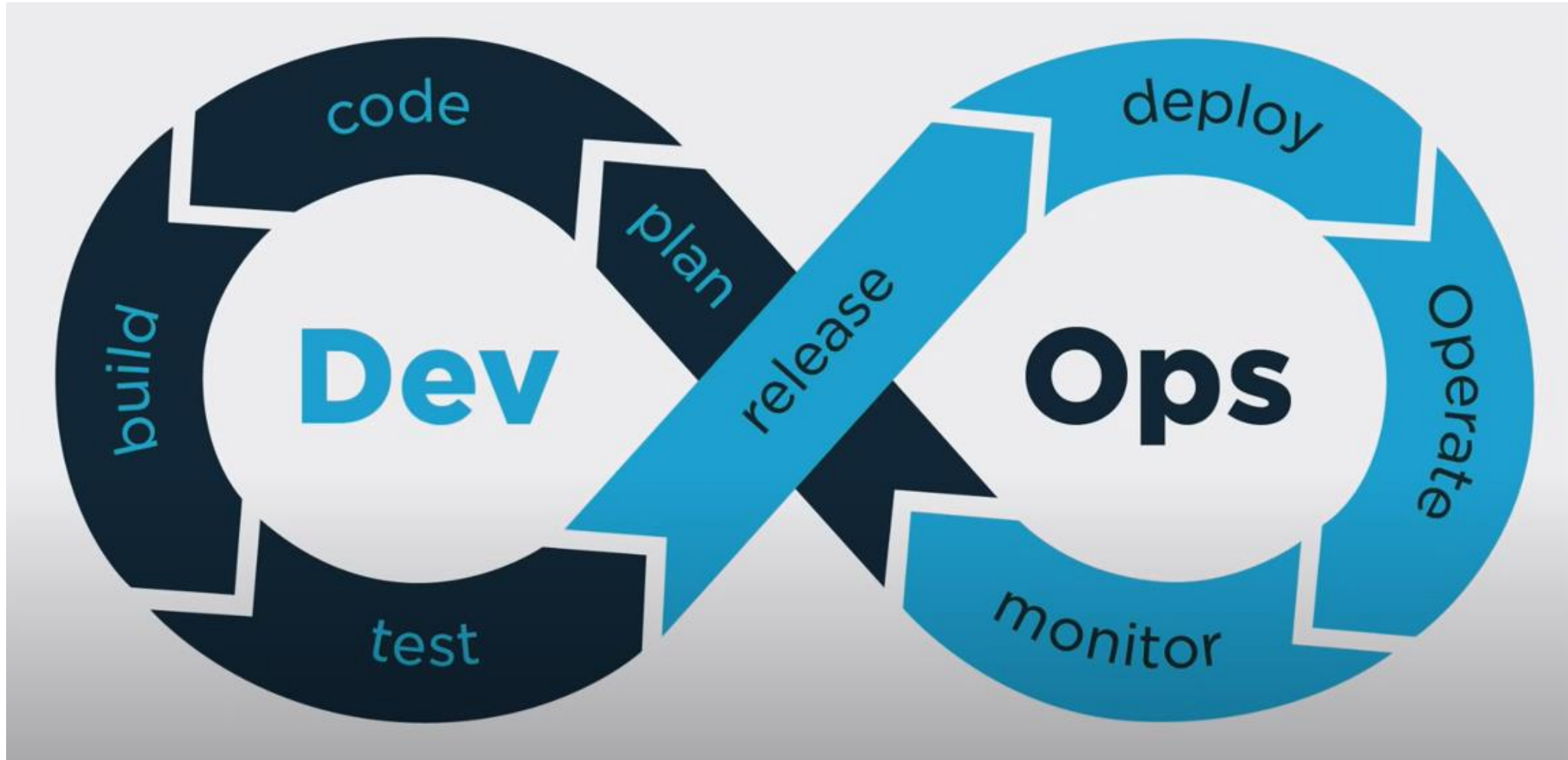
Летняя школа DevOps и CI/CD

CI/CD concept

GitLab CI/CD

- Continuous Integration / Continuous Deployment
- GitLab CI/CD
- Настройка заданий
- Настройка pipelines
- Использование контейнеров
- Основные полезные ключевые слова
- Создание первого конвейера
 - От сборки до выгрузки
- Артефакты, окружение, кэш и т.п.

Автоматизация рутинных операций



GitLab CI/CD

- Каждый продукт проходит цикл
 - Идея – Разработка (Кодинг) - Сборка - Тестирование - Упаковка – Развертывание
- Кроме того, постоянная доработка
 - Новые функции, новые версии продукта
 - Исправление багов
- Каждое изменение запускает цикл заново
- Без автоматизации процессов быстрые циклы невозможны

Что дает?

- С точки зрения разработчика
 - Он сразу видит результат и получает быстрый feedback в случае ошибок
 - Позволяет обнаруживать и исправлять ошибки в короткое время
 - Только это работает, если он код покрыт тестами
- С точки зрения работы с чужим кодом
 - Работать с покрытым тестами кодом, который снабжен скриптами для развертывания и упакован в стандартный контейнер, значительно приятнее

Что дает?

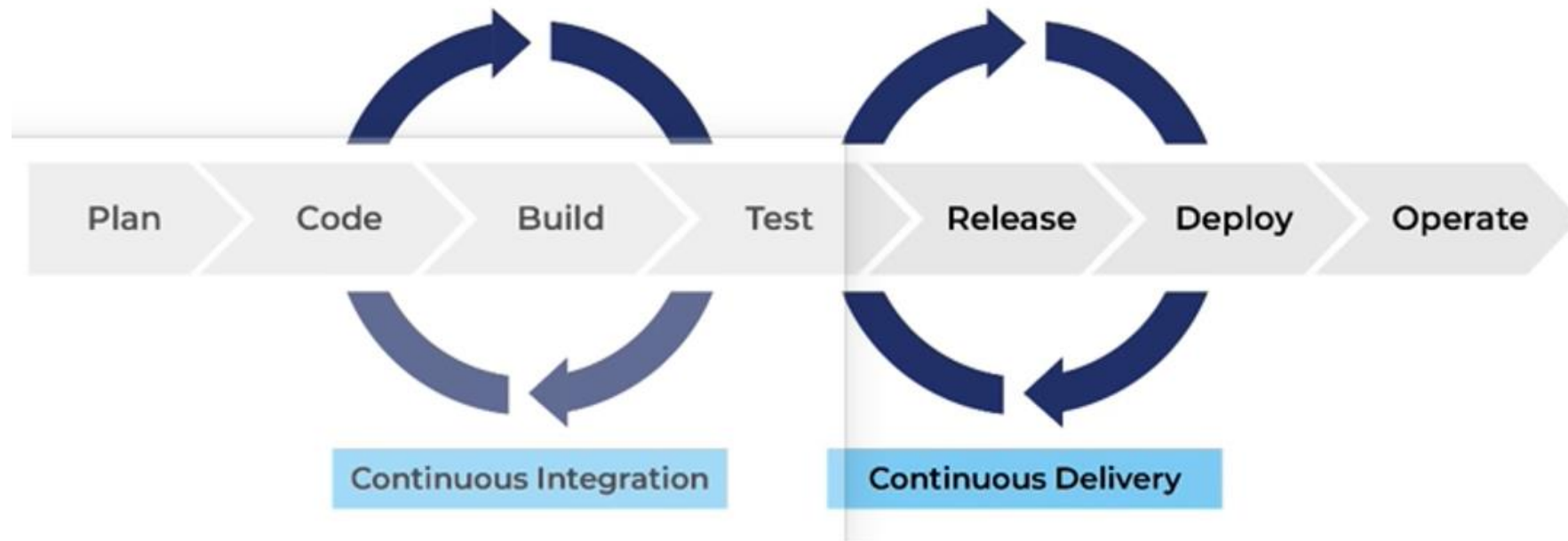
- С точки зрения человека, отвечающего за deploy
 - (системный администратор, release manager, operations project manager)
 - Теперь это DevOps инженер
 - Все базовые операции автоматизированы, остается только мониторить процесс и автоматически накатывать или откатывать изменения

Что дает?

- С точки зрения product owner
 - получает возможность быстрее оценить результат, внести изменения в задание, контролировать весь процесс
- С точки зрения менеджера проекта
 - После разработки остается не просто программный код, а готовый конвейер для тестирования, сборки и развертывания

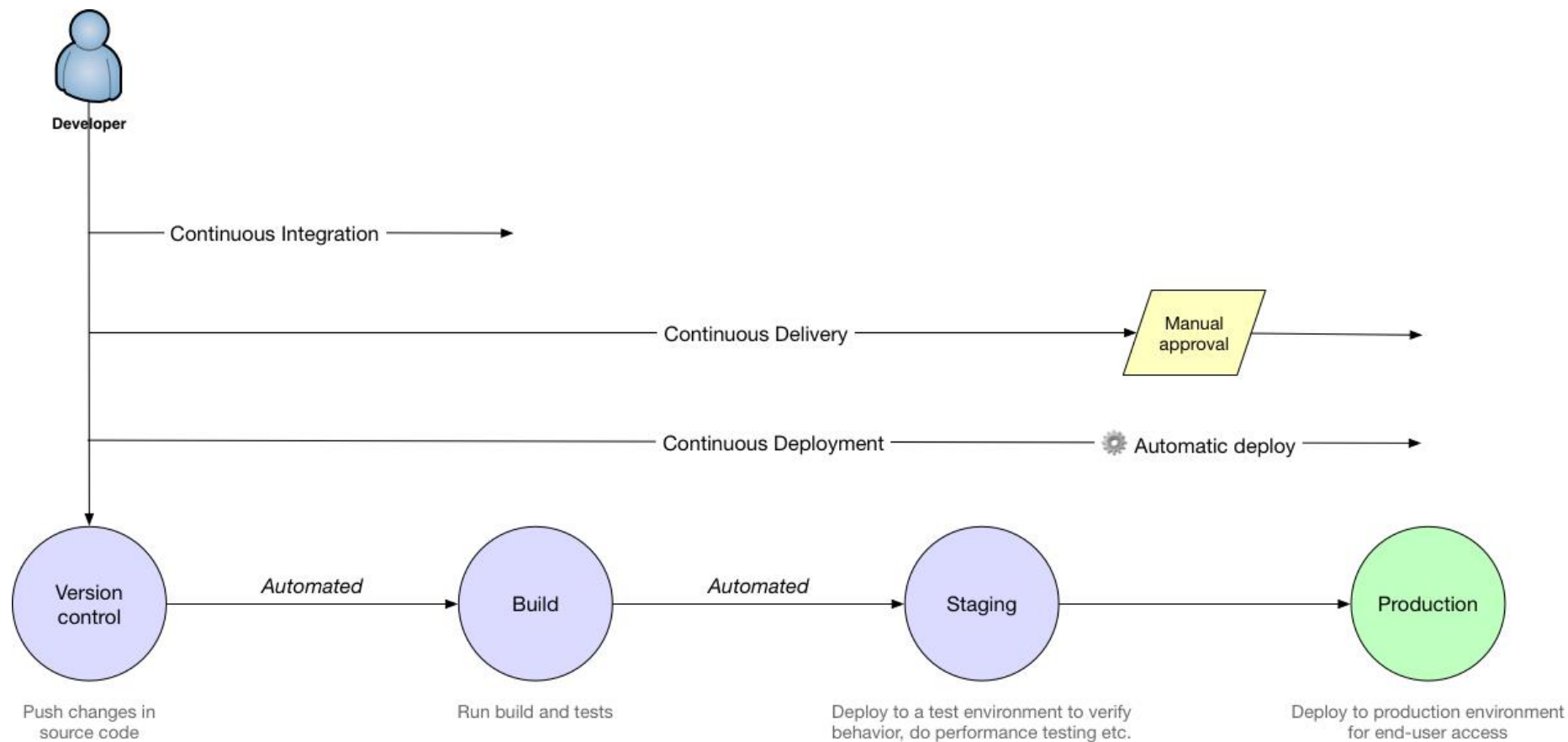
CI/CD

- Continuous integration (CI) – непрерывная интеграция
- Continuous delivery (CD) – непрерывная доставка
- Continuous deployment (CD) – непрерывное развертывание



CI/CD

- Continuous delivery / continuous deployment



Continuous integration

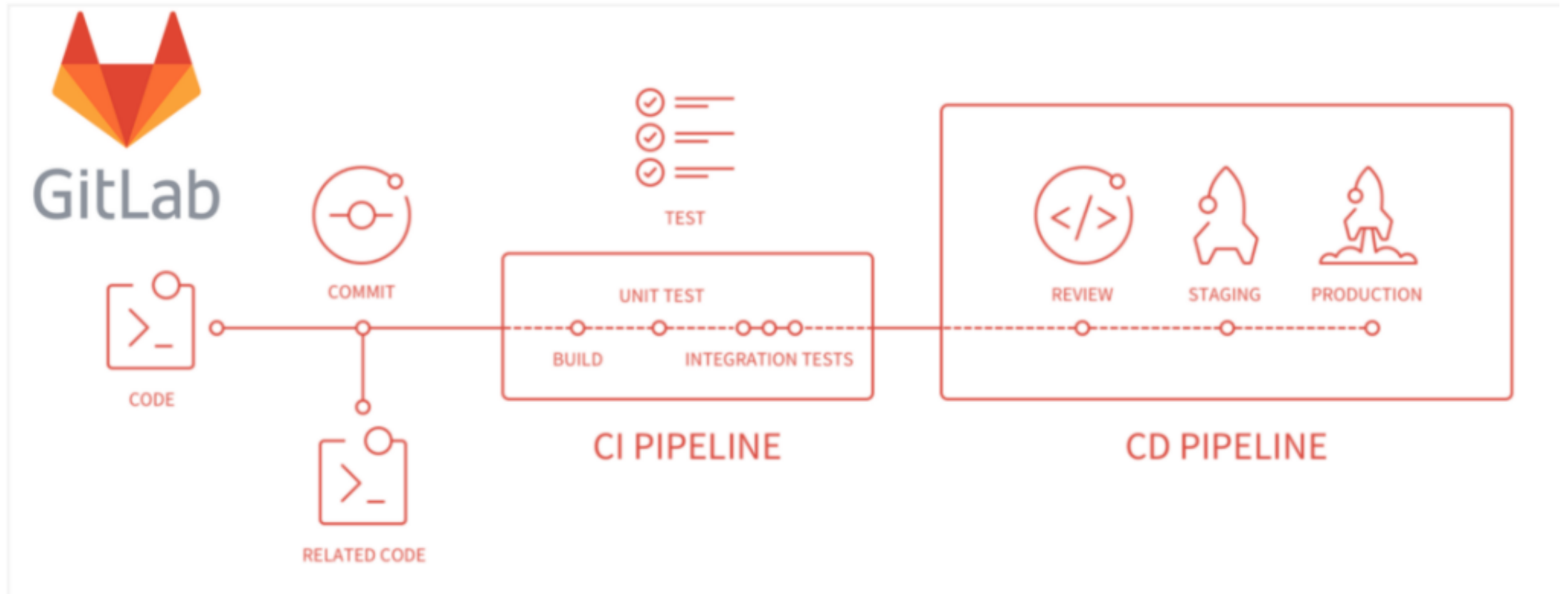
- Этапы разработки, сборки тестирования, упаковки
- Работы над кодом
 - Исправления кода, code review и т.п.
- Автотесты
 - Сюда входит этапы модульного и интеграционного тестирования
- Упаковка
 - Создание артефактов, Сборка контейнеров, заливка их в реестр
- Основная задача – это прогон автотестов и упаковка приложения (подготовка к доставке)
 - После данного этапа приложение теоретически готово к внедрению

Continuous deployment

- Этап доставки на сервер уже готового протестированного продукта
 - К моменту deploy мы уверены, что система в консистентном состоянии и все необходимые тесты прошли
- Может включать в себя настройку инфраструктуры, доставку сборки на сервер и разворачивание на сервере
- Можно настраивать автоматически, можно вручную
- Основная задача – развернуть обновленную систему в заданном окружении

CI/CD pipelines

- Deploy может идти в разные environments



Staging server

- Кроме автотестов, бывает еще ручное тестирование
 - Приемочное тестирование, канареечное тестирование и т.п.
- Для этого надо где-то развернуть приложение
 - Обычно будет несколько environments: test, staging, production
- Continuous Delivery может включать в себя deploy на test, staging, production сервер
- Для всех окружений могут быть свои правила
 - например на staging делать deploy автоматом а на production вручную и т.п.