



CI/CD

Лекторы:

Аспирант МФТИ, Шер Артём Владимирович

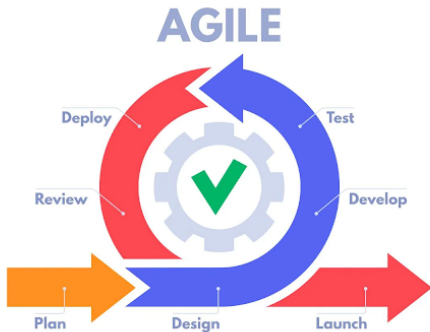
Аспирант МФТИ, Зингеренко Михаил Владимирович

19 ноября 2024

Agile

AGILE - это методология управления проектами и разработки программного обеспечения, ориентированная на гибкость и частые корректировки. В прошлом не было такой явной потребности по нескольким причинам:

- Малые масштабы проектов
- Медленная скорость разработки
- Ограниченные технологии и инструменты
- Отсутствие востребованности
- Сложность настройки автоматизации



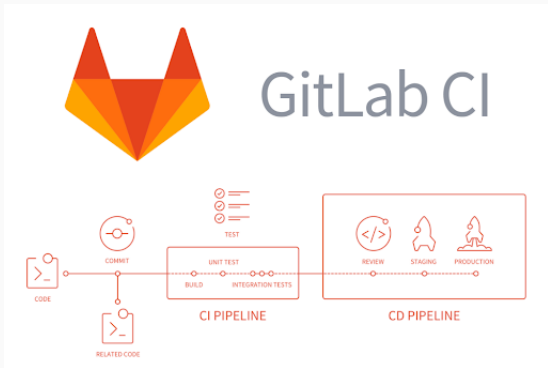
Основные принципы включают в себя

- Коллективную работу,
- Быстрые итерации,
- Постоянное взаимодействие с заказчиком,
- Способность быстро адаптироваться к изменениям в процессе разработки.

AGILE призван повысить отзывчивость и эффективность разработки, сокращая временные рамки и улучшая качество конечного продукта.

Continuous Integration

CI - это методология разработки и набор практик, при которых в код вносятся небольшие изменения с частыми коммитами.



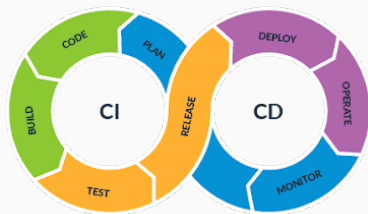
Цель CI — обеспечить последовательный и автоматизированный способ сборки, упаковки и тестирования приложений.

- Каждая интеграция включает в себя
- Автоматическую сборку
- Автоматические проверки
- Автоматические тесты (чаще всего UNIT)

Continuous Delivery

Continuous Delivery– процесс разворачивания собранного кода в среду, приближенной к production, и прохождения автоматизированных тестов для проверки готовности сборки к релизу.

- репликация production среды
- интеграционные тесты

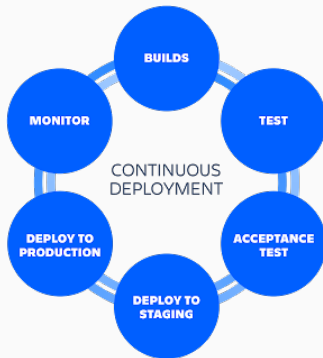


Continuous Deployment

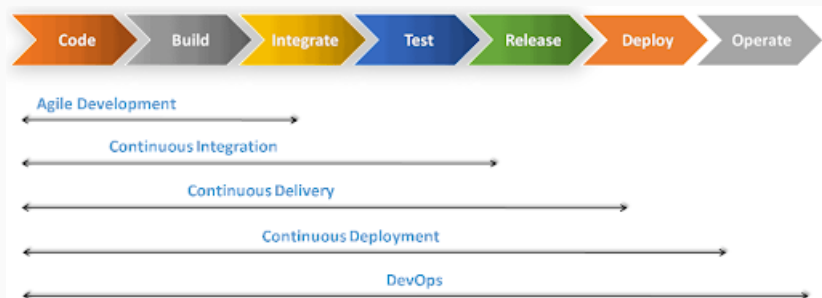
Continuous Deployment – процесс автоматической выкатки новых версий программ в production.

- Обновление программ на prod-серверах на новую версию
- Желательно, без downtime – rolling upgrade (постепенное обновление)

В результате имеем конвейер по обновленной версии ПО – CI CD pipeline



Сравнение CI/CD



Инструменты CI/CD

- Jenkins
- GitHub Actions
- Travis CI
- Drone CI
- Azure DevOps
- TeamCity
- GitLab CI/CD



Jenkins - это открытое программное обеспечение для автоматизации процессов сборки, тестирования и развертывания программного обеспечения. Цель использования Jenkins:

- Упрощение и автоматизация рутинных задач в процессе разработки, что позволяет командам разработчиков сосредотачиваться на кодировании, а не на рутинных операциях.



Jenkins

GitHub Actions

- GitHub Actions интегрирован напрямую с репозиториями на GitHub и предоставляет простой способ настройки автоматизированных процессов в рамках самого GitHub.
- В GitHub Actions используется YAML для определения рабочих процессов.
- GitHub Actions предоставляет простой способ определения и управления процессами в рамках GitHub, что делает его более простым в использовании для простых сценариев.



Travis CI и Drone CI

Travis CI - это облачный сервис непрерывной интеграции, который автоматизирует процессы сборки и тестирования приложений. Он интегрируется с репозиториями на платформе GitHub и Bitbucket. Travis CI использует файл конфигурации в формате YAML для определения шагов сборки, тестирования и развертывания приложения.

Drone CI - это также инструмент непрерывной интеграции, который позволяет автоматизировать сборку, тестирование и развертывание кода.



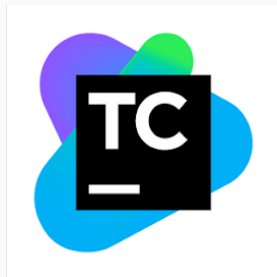
Travis CI

Azure DevOps - это полный набор инструментов и служб от Microsoft для автоматизации процессов разработки, тестирования и развертывания приложений. Он предоставляет средства для реализации практик CI/CD, а также для управления задачами, кодом, тестированием и отслеживания работ.



TeamCity - это интегрированная система CI/CD, разработанная компанией JetBrains.

- Имеет интуитивно понятный веб-интерфейс, что делает процесс настройки и управления проектами более удобным.
- Имеет интуитивно понятный веб-интерфейс, что делает процесс настройки и управления проектами более удобным.
- Поддерживает распределенные сборки, что ускоряет процесс сборки проектов.
- Имеет интеграцию с популярными средами разработки, такими как IntelliJ IDEA, Eclipse, и Visual Studio.



До следующей лекции!