SolarLab>_

Использование CI/CD систем

Содержание

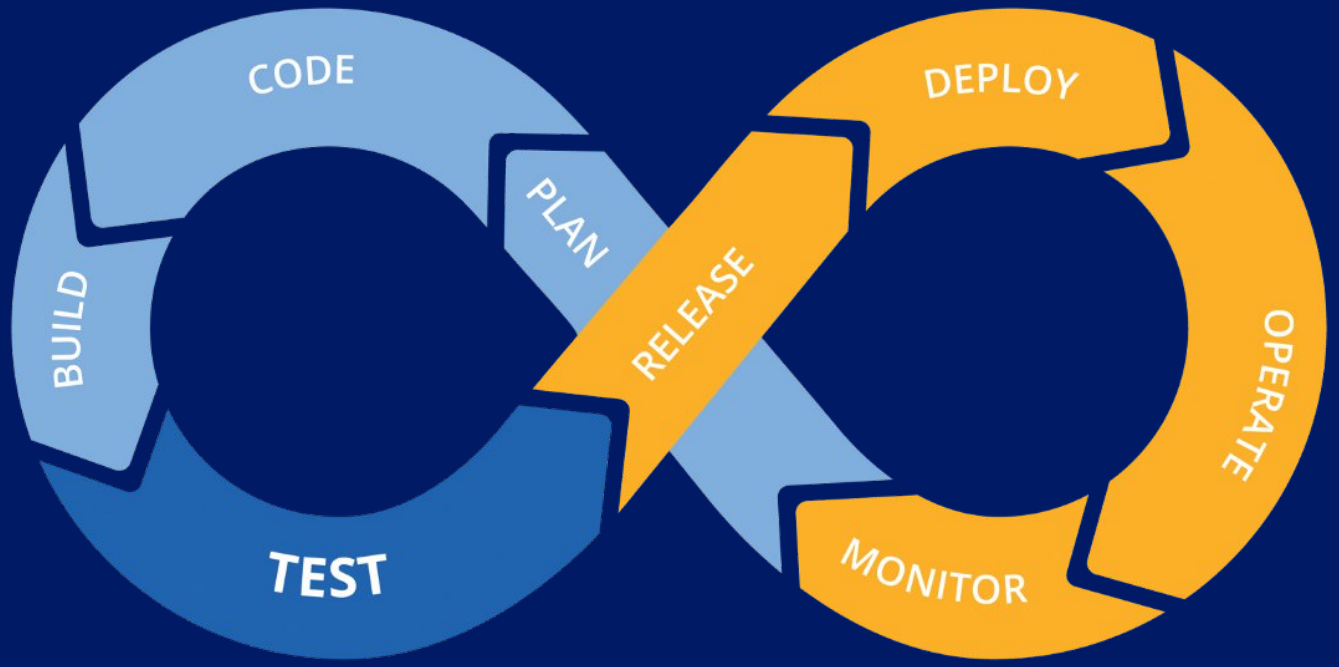
SolarLab>_

Как мы доставляем продукт конечному пользователю?

Руками? Скрипты? Что ещё?

SolarLab>_

Концепция непрерывной интеграции и доставки (CI/CD)



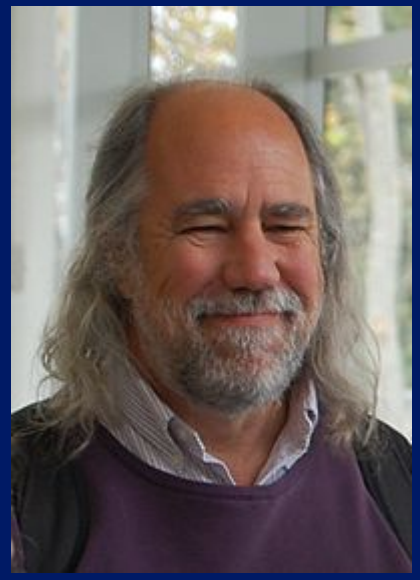
Концепция, которая реализуется как конвейер, облегчая слияние только что закомиченного кода в основную кодовую базу. Концепция позволяет запускать различные типы тестов на каждом этапе (выполнение **интеграционного** аспекта) и завершать его запуском с развертыванием закомиченного кода в фактический продукт, который видят конечные пользователи (выполнение **доставки**)

SolarLab>_

Немного истории

?

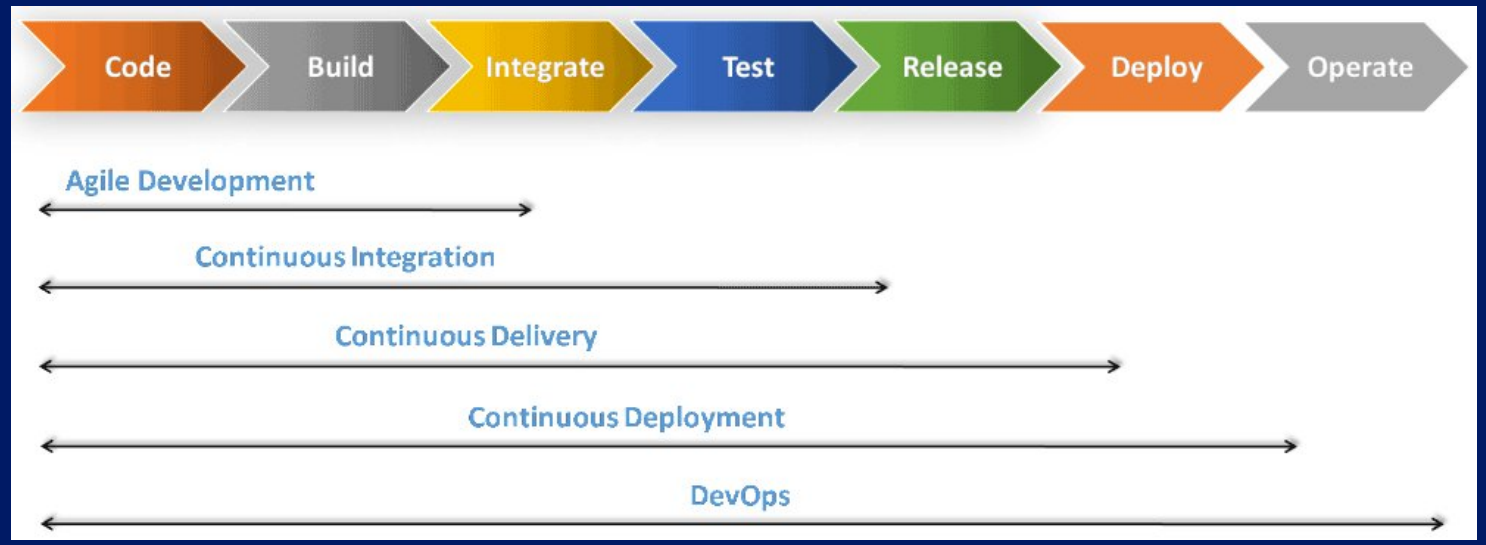
Что такое CI/CD?



Непрерывное (несколько раз в день) слияние рабочих копий программного кода в общую основную ветвь и тестирование результатов оформилось в «концепцию непрерывной интеграции и доставки софта» (англ. Continuous Integration & Continuous Delivery или сокращённо CI/CD). Впервые её предложил Гради Буч в 1991 году.

SolarLab>_

Принципы CI/CD



Принципы CI/CD

Первый принцип CI/CD: сегрегация ответственности заинтересованных сторон.

Второй принцип CI/CD: снижение риска.

Третий принцип CI/CD: короткий цикл обратной связи.

Первый принцип CI/CD: сегрегация ответственности заинтересованных сторон

- Одним из основных преимуществ CI/CD является своевременное участие различных заинтересованных сторон в любом проекте.
- **Разработчики и дизайнеры (Devs)** создают опыт и логику продукта, представленные в рассказах пользователей, и несут ответственность за создание работающий функций, доказывая это через модульные тесты.
- **Инженеры по качеству (QE)** отвечают за поддержание качества продукции, просмотр функций по мере их выполнения (см. определение «Готово») и вводят сквозные (E2E) функции / приемочные тесты, чтобы установить, что клиент получит продукт работающий без ошибок.
- **Бизнес-аналитики (BAs) и владельцы продуктов (POs)** несут ответственность за приемлемость (например, стоимость бизнеса), что подтверждается взаимодействием с фактическими пользователями и созданием пользовательских историй. Они координируют и анализируют результаты приемочных тестов для проверки рассказов пользователей и, возможно, создания новых, основанных на отзывах.
- **Оперативный отдел (Ops)/ DevOps-инженеры** несут ответственность за доступность продукта пользователям. Работая в области CI/CD, они масштабируют концепции по мере необходимости и разрабатывают логистику кода, чтобы код от разработчиков мог перейти в производственную среду и пользователи могли получать доступ.
- **Пользователи** несут ответственность за использование продукта. Подразумевается, что продукт должен быть полезен и оправдывать усилия.
- Каждый этап конвейерной обработки CI/CD создает среду, настроенную на то, чтобы ***группы брали ответственность за соответствующую стадию тестирования***, обеспечивая целостность процесса.

Второй принцип CI/CD: снижение риска

- Каждый этап конвейерной обработки CI/CD создается для снижения риска в определенном аспекте. Разработчики отвечают за логические и письменные тесты, чтобы снизить риск нарушения логики. QE отвечают за целостность потока пользователей и записывают тесты для снижения риска сломанных потоков/ историй пользователей. BAs и POs отвечают за удобство использования и принимают участие в приемочных тестах пользователей, чтобы снизить риск создания непригодных / нежелательных функций. Ops/ DevOps несут участвуют в обслуживании CI/CD, связанные с развертыванием операций (анализ схемы данных/ миграции данных) и масштабирование, чтобы снизить риск недоступности продукта.

- Поскольку этапы предназначены для снижения конкретного риска, важно отметить, что при настройке конвейерной обработки каждый этап должен также служить шлюзом для контроля качества, который предотвращает прохождение поврежденных/ нежелательных функций.

Третий принцип CI/CD: короткий цикл обратной связи

- Причина внедрения конвейерной обработки CI/CD — использование машин для работы с людьми. Это позволяет сократить время, затрачиваемое на обратную связь по разрабатываемым функциям.
- Но почему выгоднее использовать машины? Потому что ***люди не масштабируются, как машины.*** С помощью масштабирования сокращается время, затрачиваемое на тестирование программного обеспечения, что позволяет автоматизировать процесс развертывания. Эти процедуры займут гораздо больше времени, если будут выполняться человеком.
- Однако в ошибкоопасных ситуациях, требующих ввода данных человеком, автоматизация может быть нежелательной/ невозможной. Например, мы никогда не сможем автоматизировать тестер, когда дело доходит до удобства использования. Еще одним важным примером является миграция производственных данных. Автоматизируйте логистику кода (как код упакован/ перемещается между этапами), насколько это возможно, но имейте в виду: автоматизация в конечном итоге предназначена для ***сокращения времени, затрачиваемого на получение закомиченного кода***. Если присутствуют ошибки, которые требуют больше времени для исправления, чем для предотвращения, избегайте автоматизации и старайтесь достичь короткого цикла обратной связи.

SolarLab>_

Инструменты для CI/CD

Инструменты для CI/CD

Локальные

GitLab CI

TeamCity

Bamboo

GoCD Jenkins

Circle CI

и другие…

Инструменты для CI/CD

Облачные

BitBucket Pipelines

Heroku CI

Travis

Codeship

Buddy CI

AWS CodeBuild

и другие…

Инструменты для CI/CD

Правительственные

Hats

Nectar

и другие…

SolarLab>_

Наши проекты

На наших проектах мы используем

TeamCity

Jenkins

Jenkins

Вопросы?

Ссылки

- **That CI/CD Thingy: Principles, Implementation & Tools** https://blog.gds-gov.tech/that-ci-cd-thing-principles-implementation-tools-aa8e77f9a350
- **Непрерывная интеграция Wiki** https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F
- **Непрерывная интеграция и доставка (CI/CD): идеальная методика разработки или отраслевой хайп?** https://tproger.ru/blogs/ci-cd/
- **От репозитория до CI/CD-инфраструктуры в продакшене за неделю** https://habr.com/ru/company/itsumma/blog/333650/

SolarLab>_

Спасибо за внимание