

Практика Continuous Integration

Введение > CI > Среды > CD > CT > Метрики

Непрерывная интеграция (CI, Continuous Integration) – это практика разработки программного обеспечения, которая заключается в выполнении частых автоматизированных сборок для скорейшего выявления ошибок. Это позволяет постоянно быть уверенным в том, что код находится в рабочем состоянии. Переход к непрерывной интеграции позволяет снизить трудоёмкость интеграции изменений кода с существующей кодобазой и сделать её более предсказуемой за счет наиболее раннего обнаружения и устранения ошибок и противоречий, сделать процесс разработки прозрачным для всех участников команды.

Некто из ИТ: В DevOps есть практики Continuous Integration. Видимо будем менять корпоративную шину.



Практика включает в себя:

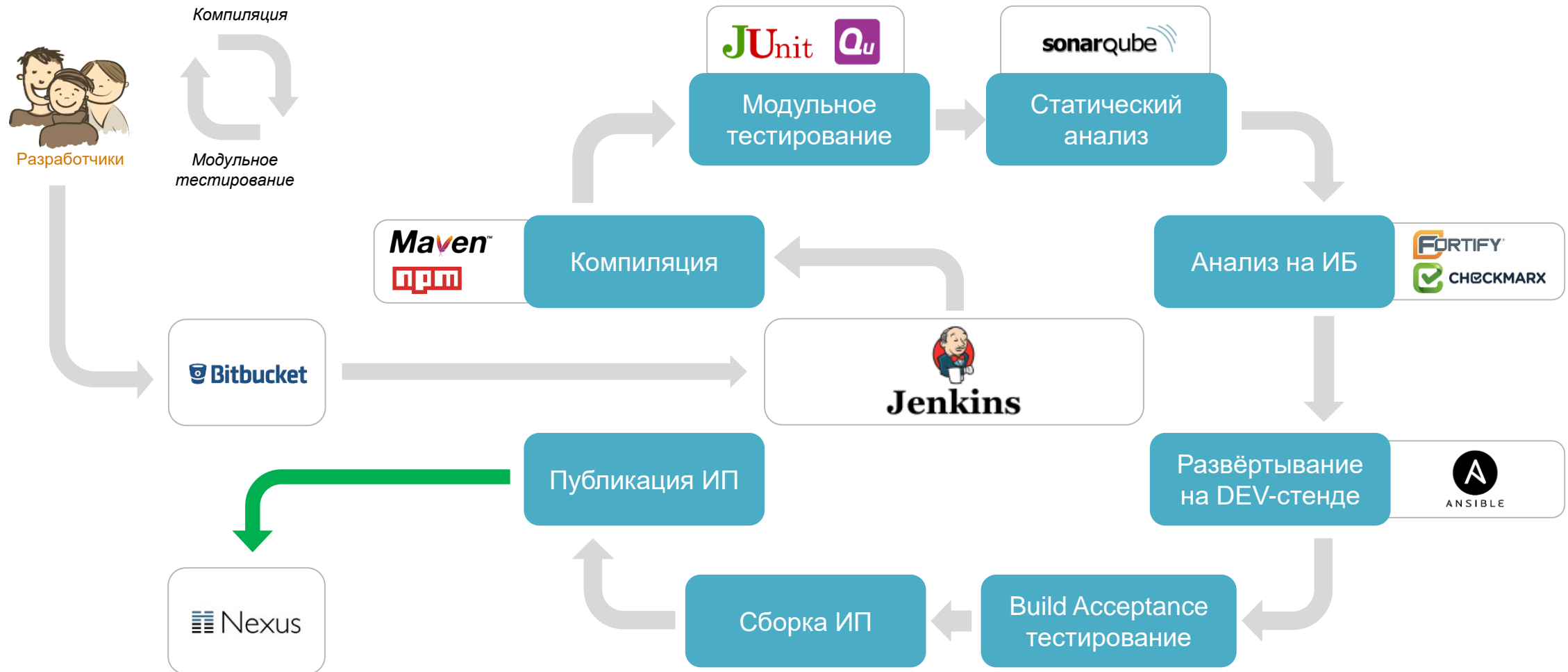
- хранение кода в системе контроля версий
- ручное code review
- частые автоматизированные сборки
- автоматизированное модульное тестирование (unit tests)
- статическую проверку качества кода
- анализ кода на безопасность*
- автоматизированное развертывание на среды разработки
- проверки работоспособности системы после развертывания (Sanity checks, Health checks, Smoke tests)
- автоматизированные Build Acceptance Tests на стендах разработки



*Подключается согласно плану подключения к программе Кибербезопасности

Continuous Integration Workflow

Введение > CI > Среды > CD > CT > Метрики



- При ошибке на любом шаге цикл прерывается с уведомлением
- В зависимости от используемых в проекте технологий порядок выполнения шагов может незначительно меняться

Инструменты CI

Введение > CI > Среды > CD > СТ > Метрики



Инструменты **модульного тестирования**, генерирующие отчёты о прохождении тестов и покрытии кода тестами в xml-формате. Например, [JUnit](#) для Java, [NUnit](#) для .Net, [QUnit](#) для JavaScript.



Сборка и управление зависимостями. Например, [Maven](#) для Java, [nuget](#) для .Net, [npm](#), [bower](#), [gulp](#) для javascript, css и статики



Статический **анализ качества кода** – [SonarQube](#)



Статический **анализ кода на безопасность** – [HPE Security Fortify Static Code Analyzer](#) или [Checkmarx](#)



Хранилище артефактов и ИП – [Sonatype Nexus OSS](#)



Оркестратор процесса CI – [Jenkins](#)

- Запуск планов сборки по событию (коммит, пул-реквест)
- Интеграция со всеми вышеописанными инструментами
- Управление потоком сборки