

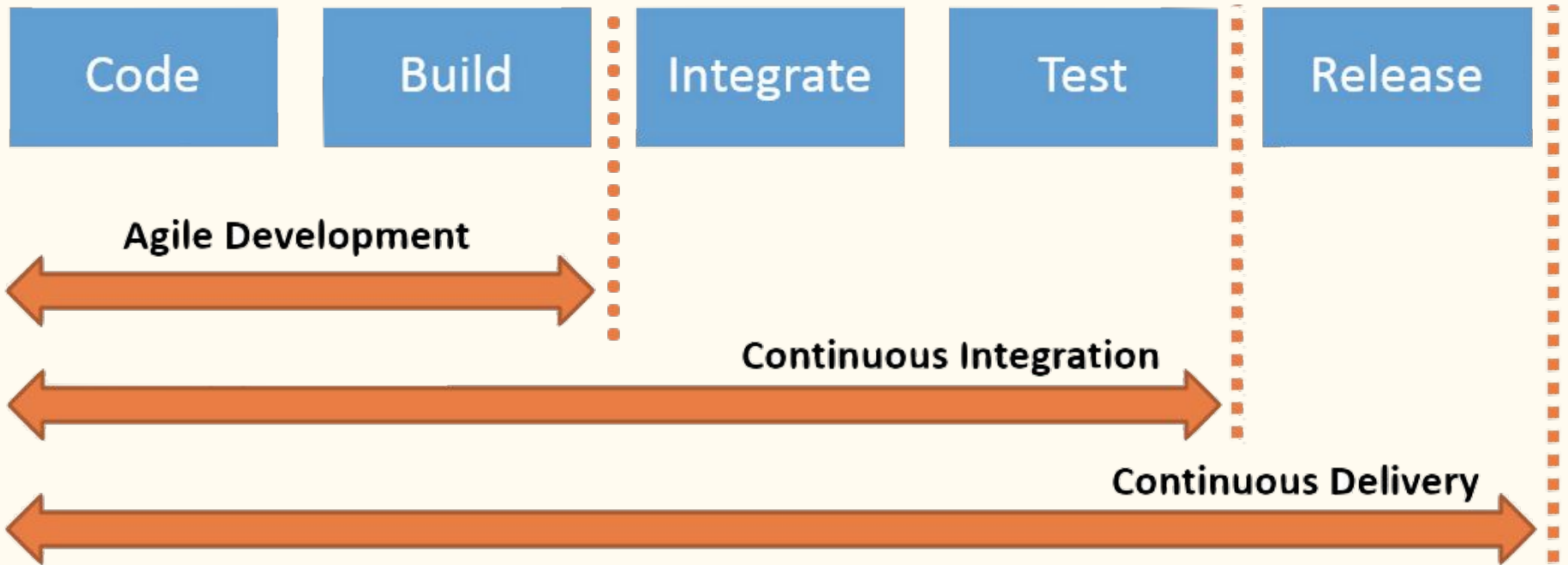
# Continuous Integration

—

Continuous Delivery

Continuous Integration подразумевает частые автоматизированные сборки проекта с целью быстрого выявления интеграционных проблем. У вас всегда будет актуальная и готовая к тестам версия продукта.

Continuous Delivery предполагает частую доставку обновлений на «боевую» систему. Продукт поддерживается в актуальной версии, а любые ошибки при обновлении проще отследить, так как при каждом релизе объем изменений невелик.



# Соглашения CI

- часто отправлять новые версии кода в репозиторий (не реже чем 1 раз в день)
- писать тесты
- запускать private builds(процесс сборки, который выполняется с использованием исходного кода, в данный момент находящегося в локальном репозитории разработчика)
- не отправлять в репозиторий неработающий код
- чинить сломанную общую ветку немедленно
- следить за тем, чтобы все тесты и проверки проходили
- не pull'ить из репозитория сломанный код
- пользоваться линтерами

# Скрипт сборки

- подготовка окружения
- rebase
- компиляция (или статический анализ кода для интерпретируемых языков)
- интеграция с базой данных
- запуск автоматических тестов
- запуск других проверок (линтеры и другие полезные утилиты)
- генерация документации
- время работы
- очистка окружения

# Автотесты

- модульные (unit)
- функциональные
- нагрузочные
- интеграционные

# Рекомендации для тестов

- 1 тест - 1 проверка (assert)
- чем быстрее тест, тем раньше он должен быть запущен
- правильное именование тестов
- группировка тестов по назначению
- на все баги должны писаться тесты
- результаты тестирования не должны зависеть от количества запусков тестов
- хорошо если тесты можно запускать параллельно

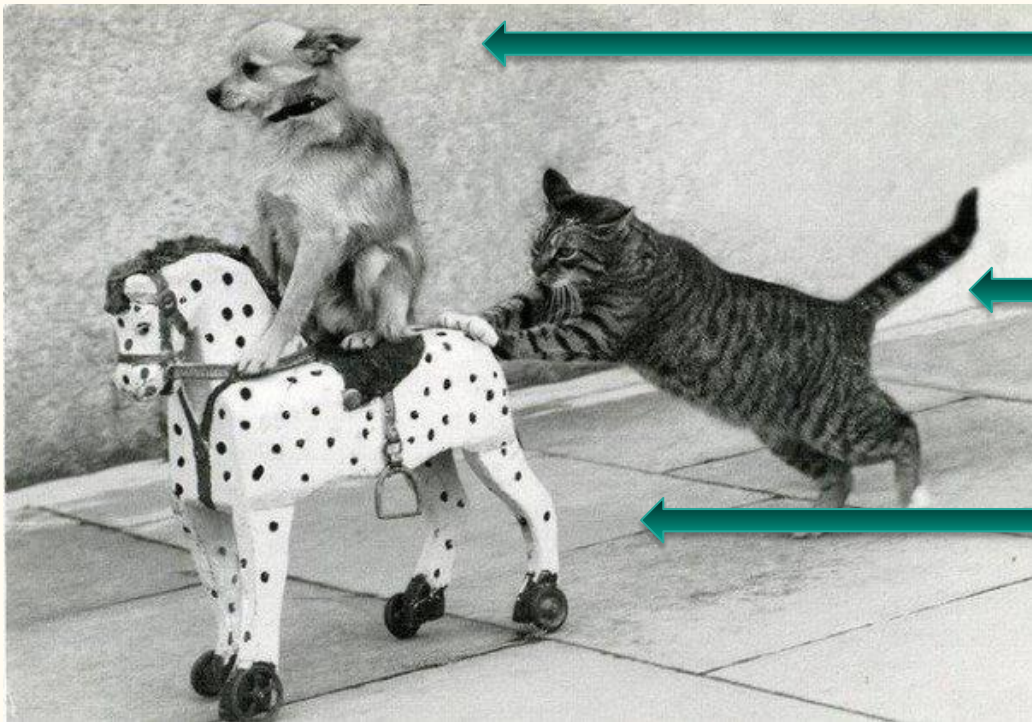


# Информирование команды

- SMS
- IM
- E-mail
- DashBoard
- Andruino

---

# Три слагаемых успеха



✓ Система аналитики

✓ Реактивный релизный менеджмент

✓ Система A/B тестирования

# Зачем все это нужно?

Гипотеза

Эксперимент

Анализ

Управляющее воздействие на бизнес

