Continuous Integration (CI)
Continuous Delivery (CD)

CI / CD

- Непрерывная интеграция, Continuous Integration (CI)
- Непрерывное развёртывание, Continuous Delivery (CD)
  - Continuous Deployment
- CI/CD – важные методы современной разработки ПО, направленные на оптимизацию процесса разработки и повышение качества ПО
- Определения CI/CD по версии Amazon

CI

- Практика, при которой изменения кода разработчиков *часто* мержатся в общий репозиторий кода
- Не нужно дожидаться, пока чьи-то изменения пройдут полный цикл интеграции в приложение
- CI как бы поощряет разработчиков сразу же интегрировать свои изменения по мере готовности кода
  - рабочие копии кода разработчика могут синхронизироваться с master-веткой несколько раз в день
- CI запускает автоматизированный процесс сборки, который компилирует код, запускает тесты и выполняет другие проверки
- Такой подход выявляет проблемы интеграции (конфликты кода, проблемы совместимости версий и т.д.) на ранних этапах, в режиме реального времени, снижает задержки релизов кода

CD

- Идёт рука об руку с CI
- Расширяет концепцию за счёт автоматизации процесса развёртывания
- Как только код проходит все проверки на этапе CI, код готов к развёртыванию, и автоматически деплоится средствами CD в различных средах (testing, prestable, stable, production)
- Основная цель CD – быстрое предоставление новых функций приложения конечным пользователям. Сюда же входит быстрый фикс багов приложения, улучшений производительности и т.д.
- Способствует тому, чтобы компании быстро «релизили» изменения в ПО, при этом поддерживая гибкие методологии разработки

Преимущества CI & CD

- CI
  - Раннее обнаружение проблем/конфликтов
  - Улучшение качества кода
  - Более быстрая обратная связь о влиянии изменений кода
  - Более тесное сотрудничество команды, прозрачность процесса разработки
- CD
  - Быстрый выход изменений на рынок, роста качества метрик (Time to market)
  - Консистентные релизы – стандартизированный список процедур, минимум конфигурационных ошибок
  - Повышенная стабильность (благодаря автоматизированным релизам и тщательному* тестированию на этапе CI вероятность ошибок снижается)
  - Возможности отката к предыдущей версии приложения. Пайплайны CD включает такие механизмы, и влияние потенциальных проблем сводится к минимуму

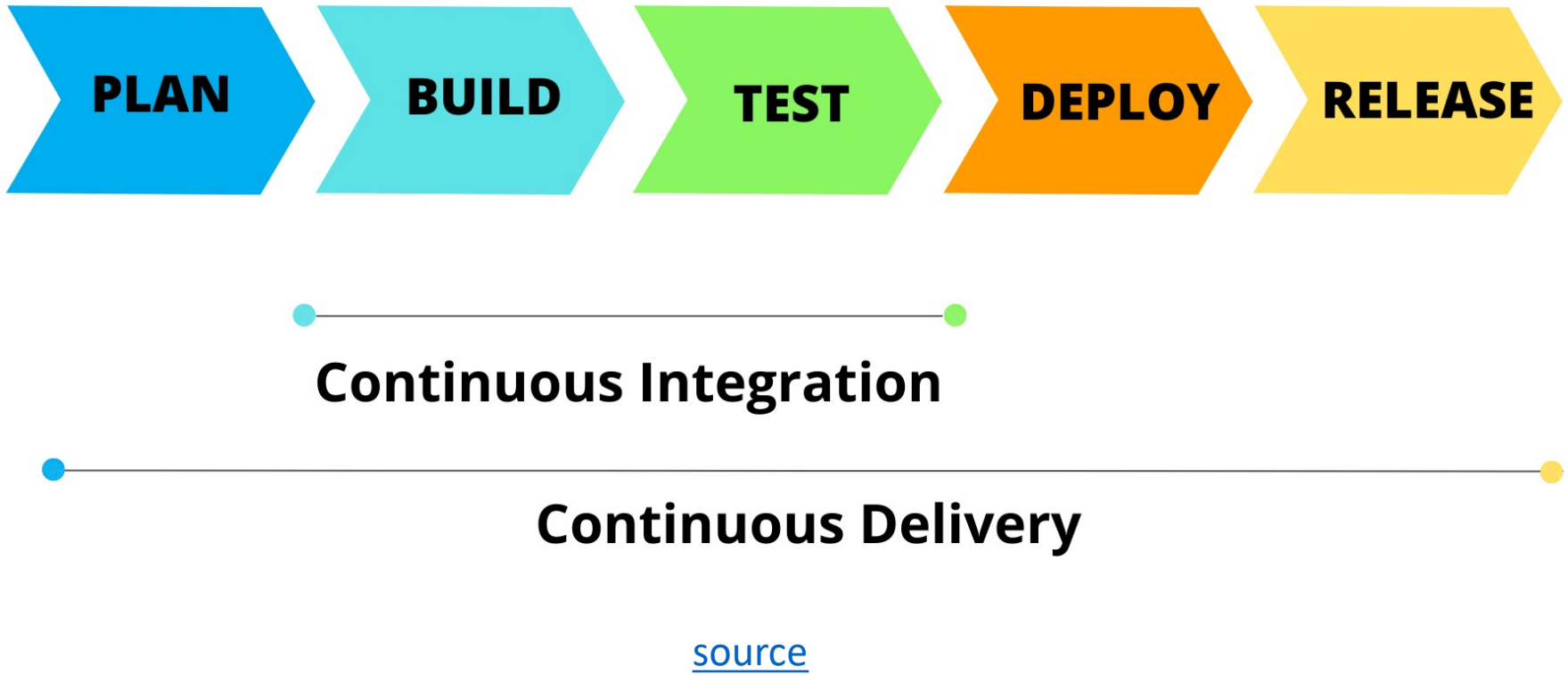
* качество тестов зависит от вас :)

Польза для разработчиков

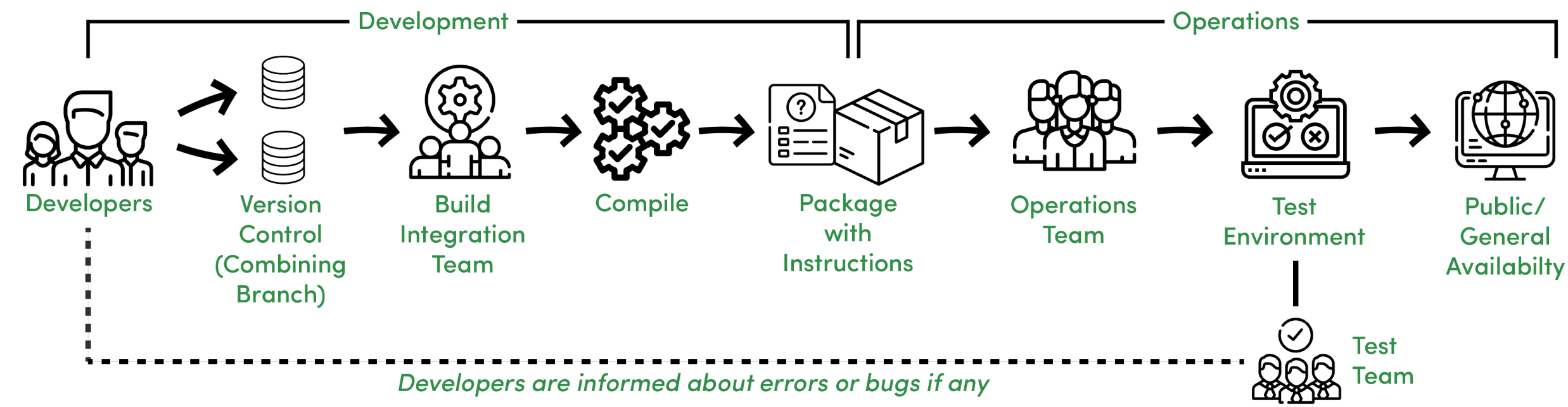
- Автоматизируя процессы интеграции, тестирования и развертывания, разработчики могут сосредоточиться на написании кода/развитии сервисов, сохраняя при этом высокое качество ПО и своевременно предоставляя ценность конечным пользователям.

razorops

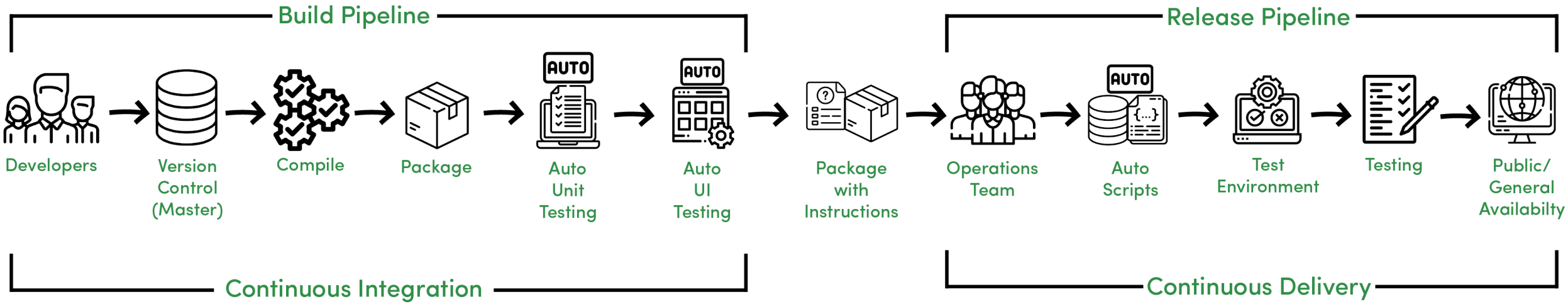
Continuous Integration and Continuous Delivery



source

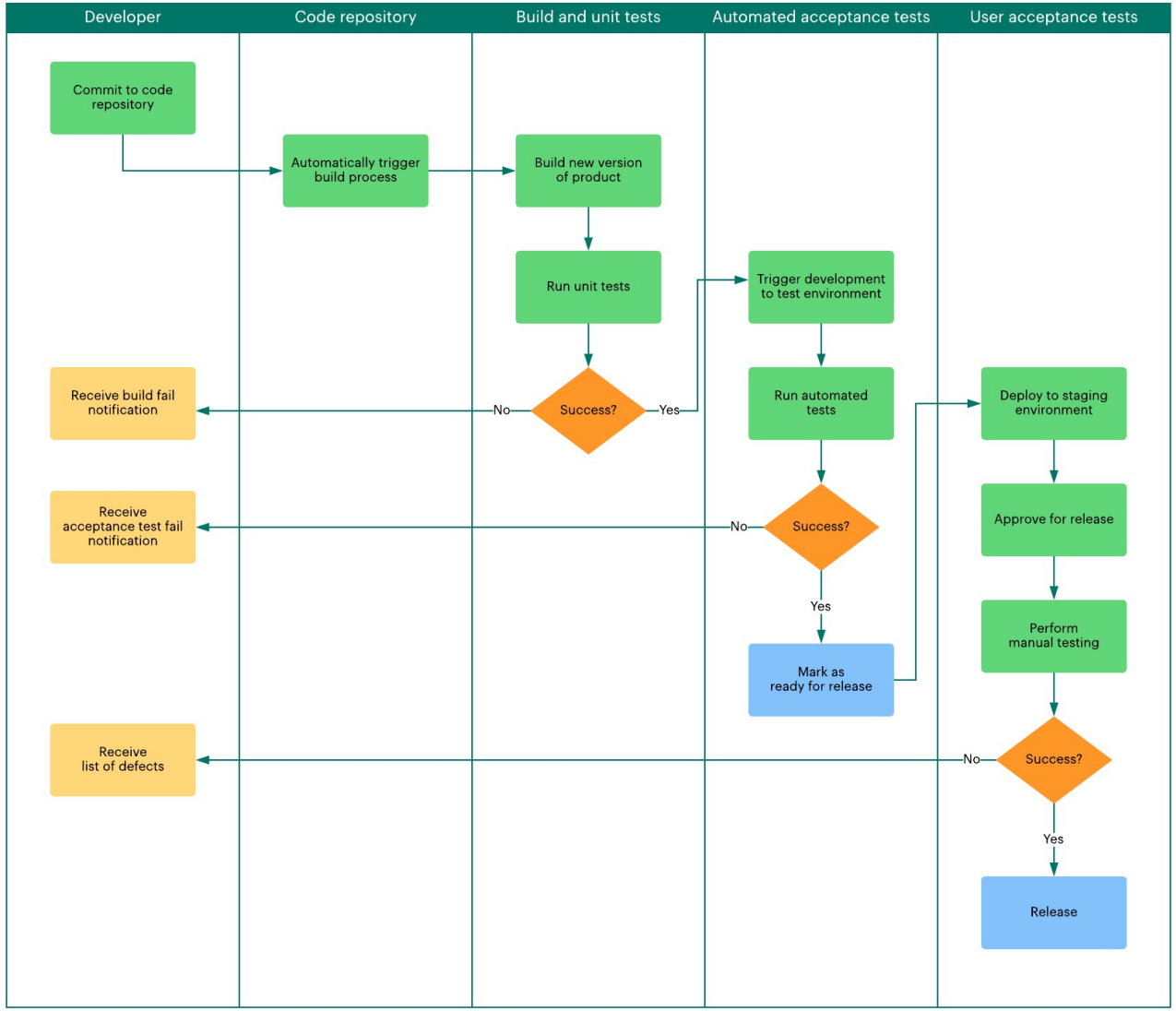


VS



images source

Continuous Deployment Process Flow



Made in Lucidchart

source

GitHub Actions

- Платформа для автоматизации процессы разработки
  - Сюда же входят CI/CD
- Ваш репозиторий меняется, и в ответ на эти изменения запускаются автоматические действия (**GitHub Actions**)
  - **GitHub Events**: новый Pull Request, добавляется новый Issue, добавляется новый контрибьютор, PR смержен в master и т.д.
- Документация

Maven Action

```yaml
name: Java CI with Maven

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - name: Checkout the repository
      uses: actions/checkout@v3
    - name: Set up JDK 1.11
      uses: actions/setup-java@v2
      with:
        java-version: '11'
        distribution: 'adopt'
    - name: Build with Maven
      run: mvn -B package --file pom.xml
```

Unit Tests Action

```yaml
name: tests
#on: push
on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]
  schedule:
    - cron: '*/2 * * * *' # every 2 minutes
  workflow_dispatch:

jobs:
  run_tests:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout the repository
        uses: actions/checkout@v3
      - name: Set up JDK 1.11
        uses: actions/setup-java@v2
        with:
          java-version: '11'
          distribution: 'adopt'
      - name: Cache Maven packages
        uses: actions/cache@v2
        with:
          path: ~/.m2
          key: ${{ runner.os }}-m2-${{ hashFiles('**/pom.xml') }}
          restore-keys: ${{ runner.os }}-m2
      - name: Run tests with Maven
        run: mvn -B test --file pom.xml
```

GitHub Actions в ваших проектах

- **TODO::**Добавить сборку и деплой в GitHub Actions

Орг. вопросы

- Расписание оставшихся занятий:
  - 10.06 в 13:00-14:20. Онлайн, доп. демо-день
  - 13.06 в 18:10-19:30 (как обычно)
  - 17.06 в 14:40-16:00. Онлайн, тема
  - 20.06 в 18:10, 2 заключительные пары. Защита проектов