

CI\CD



Алексей
Метляков



Алексей Метляков

DevOps Engineer

OpenWay



План занятия

1. [Continuous Integration](#)
2. [Test](#)
3. [Build](#)
4. [Merge](#)
5. [Continuous Delivery](#)
6. [Release](#)
7. [Test](#)
8. [Continuous Deploy](#)
9. [Deploy to Prod](#)
10. [Feedback](#)
11. [Итоги](#)
12. [Домашнее задание](#)



Continuous Integration

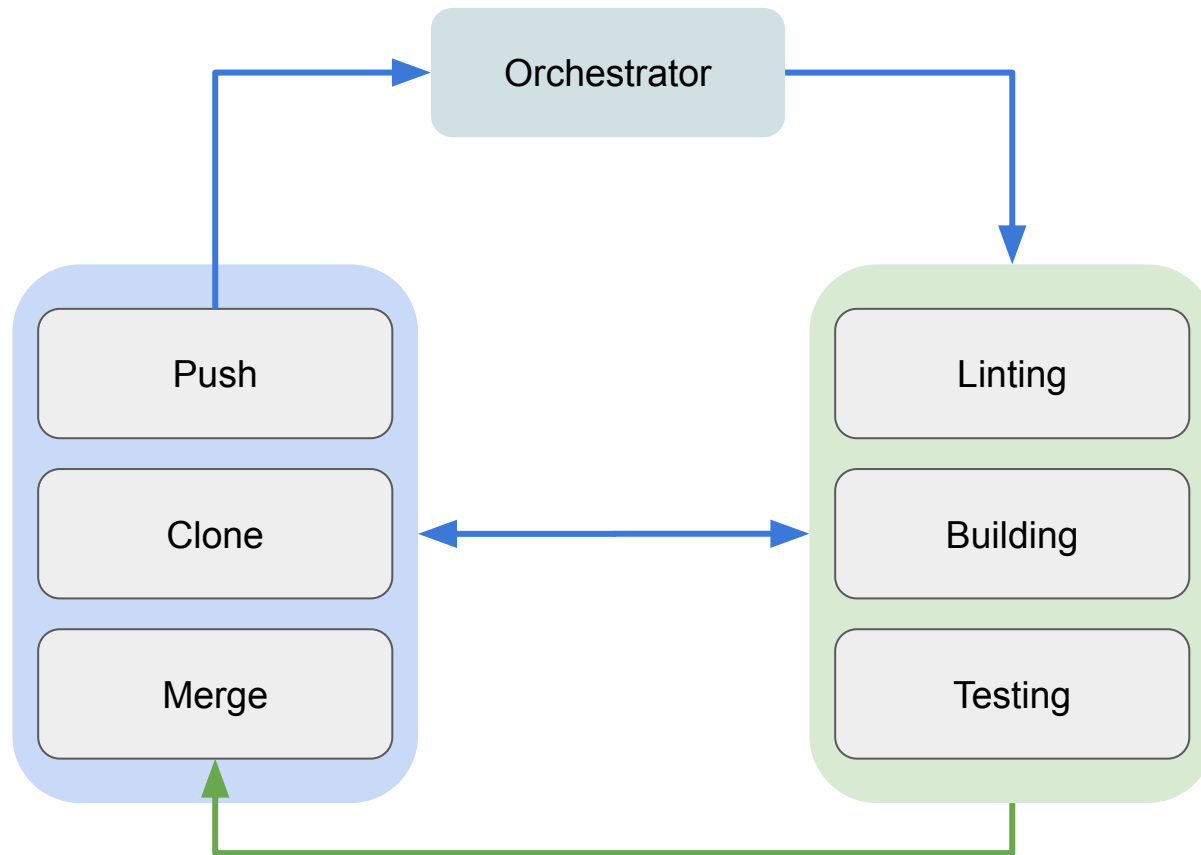
Continuous Integration

CI (Continuous Integration, непрерывная интеграция) – этап непрерывной поставки продукта, который отвечает за непрерывную **сборку** кода разработчиков.

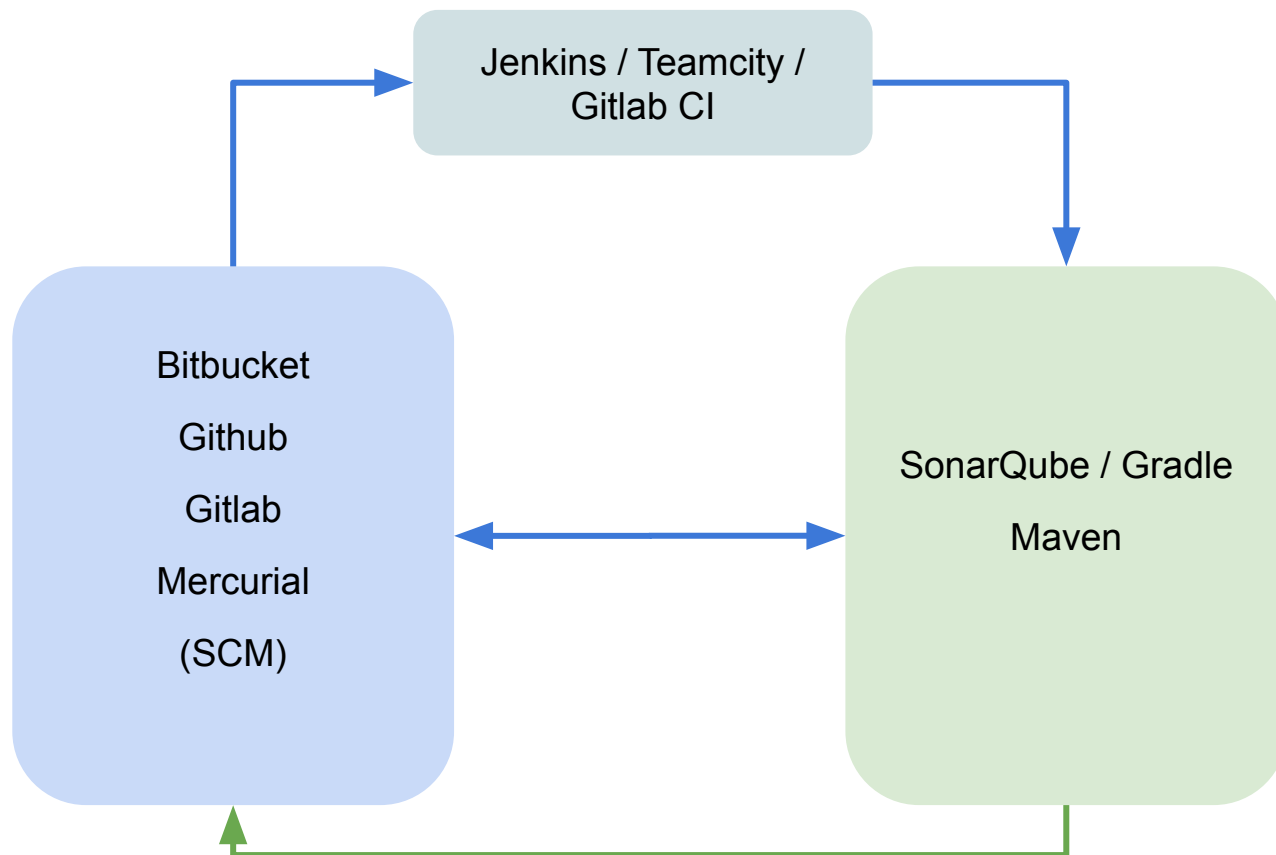
Условно, его можно разделить на следующие подэтапы:

- **Build** – сборка;
- **Test** – тестирование;
- **Merge** – слияние.

Continuous Integration



Continuous Integration



Continuous Integration

SonarQube – отдельный сервис для статического анализа кода (линтер).

Особенностью является возможность анализа **разных** языков программирования.

- Существует CE (с урезанным функционалом)
- Есть возможность купить лицензию с поддержкой

Continuous Integration

Сборщики – программы, которые позволяют:

- собрать все зависимости, необходимые для компиляции кода;
- произвести сборку кода;
- запустить unit-тесты;
- сделать тестовый запуск продукта.

Примеры сборщиков:

- Gradle
- Maven



Continuous Integration

Merge – процесс объединения текущей ветки разработчика с основной (или релизной) веткой в репозитории.



Continuous Delivery

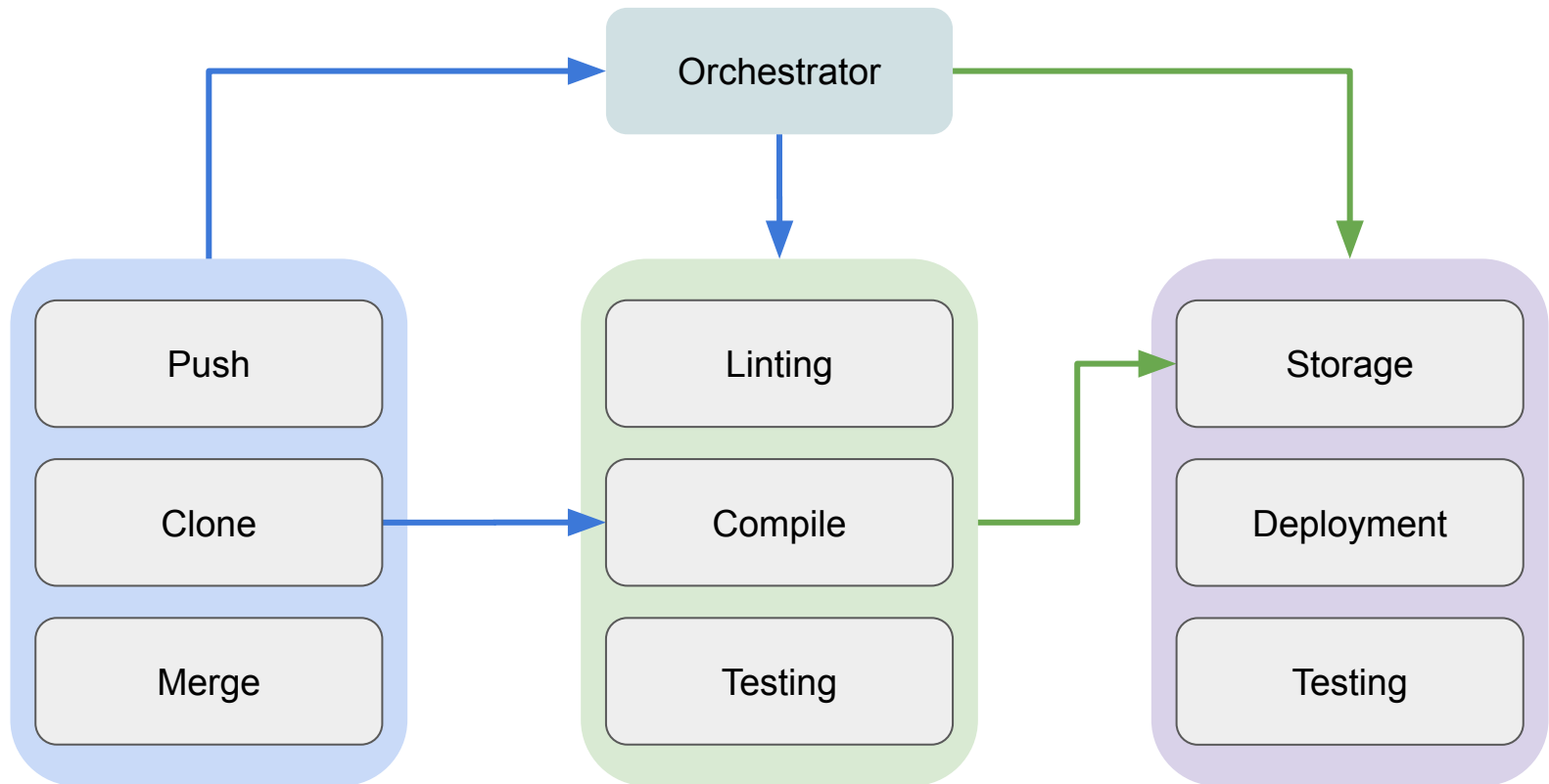
Continuous Delivery

CD (Continuous Delivery, непрерывная доставка) – этап непрерывной поставки продукта, который отвечает за **компиляцию** дистрибутива продукта, **установку** продукта на тестовые среды и проведение **тестирования**.

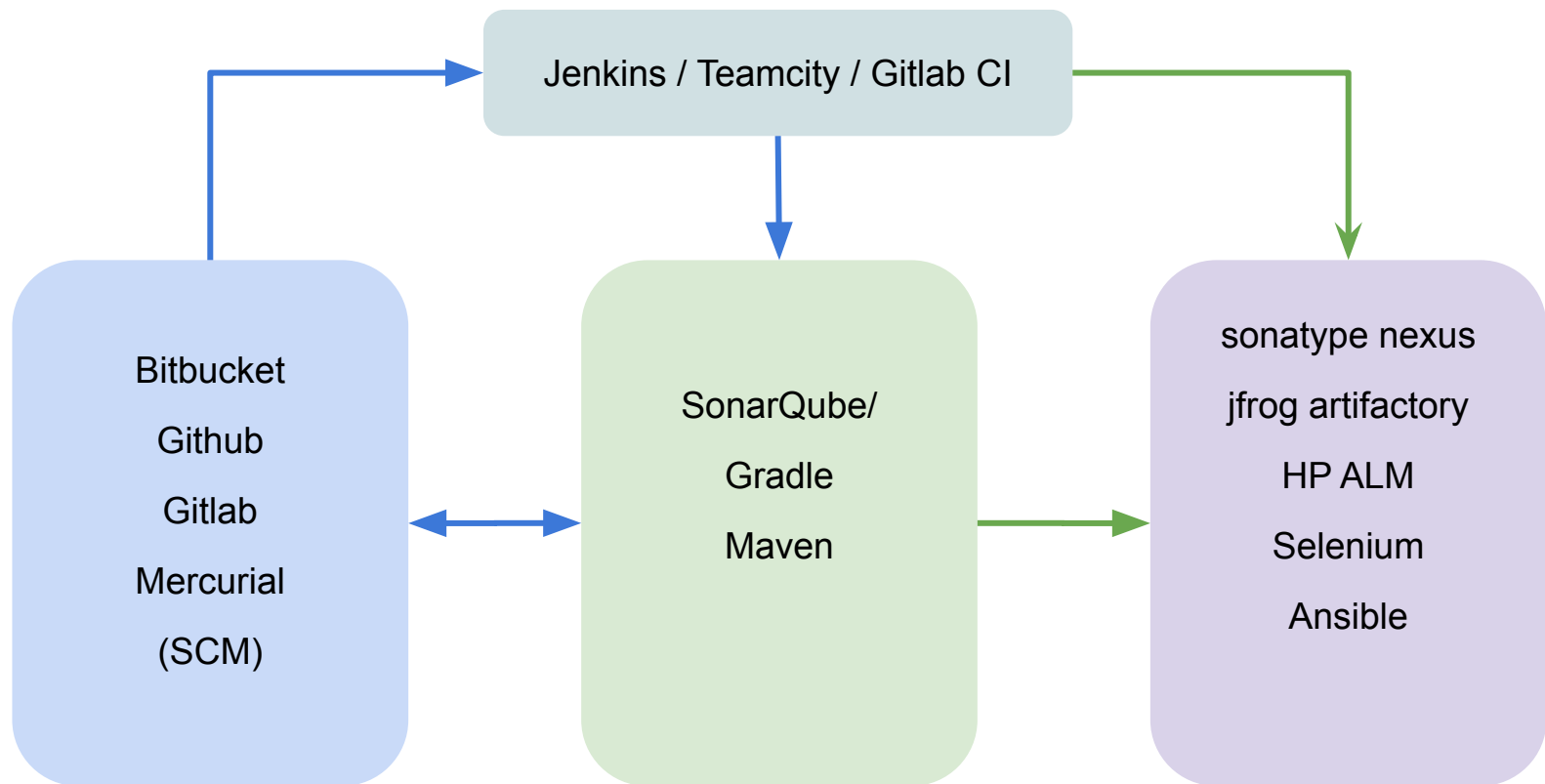
Его тоже можно поделить на подэтапы:

- **Compile** – компиляция
- **Release** – релиз
- **Deploy** – установка
- **Test** – тестирование

Continuous Delivery



Continuous Delivery



Continuous Delivery

Nexus - один из способов хранить артефакты (релизы, зависимости, библиотеки etc) и использовать их в процессах **DevOps**.

При его помощи можно:

- Хранить и версионировать артефакты
- Разделять разные виды артефактов по разным типам репозиториев
- Доставлять артефакты по всей инфраструктуре
- Использовать разные механизмы разрешения зависимостей

Continuous Delivery

Selenium - инструмент для проведения **E2E-тестирования** работоспособности web-продуктов.

При его помощи можно:

- Проводить тестирования интерфейсов
- Указывать ожидаемый результат обработки нажатий и сравнивать с полученным результатом
- Записывать ролики прохождения тестирования



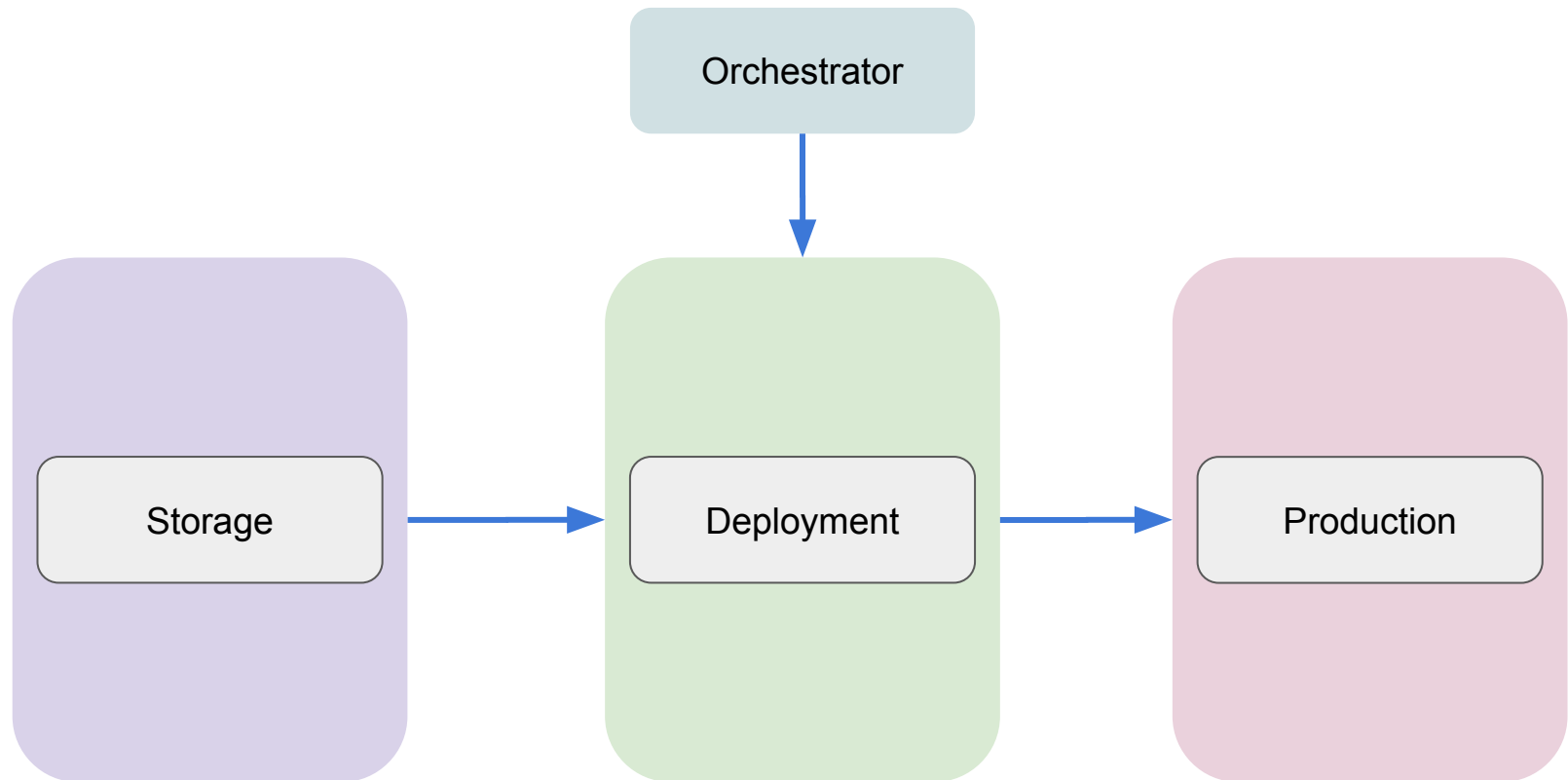
Continuous Deployment

Continuous Deployment

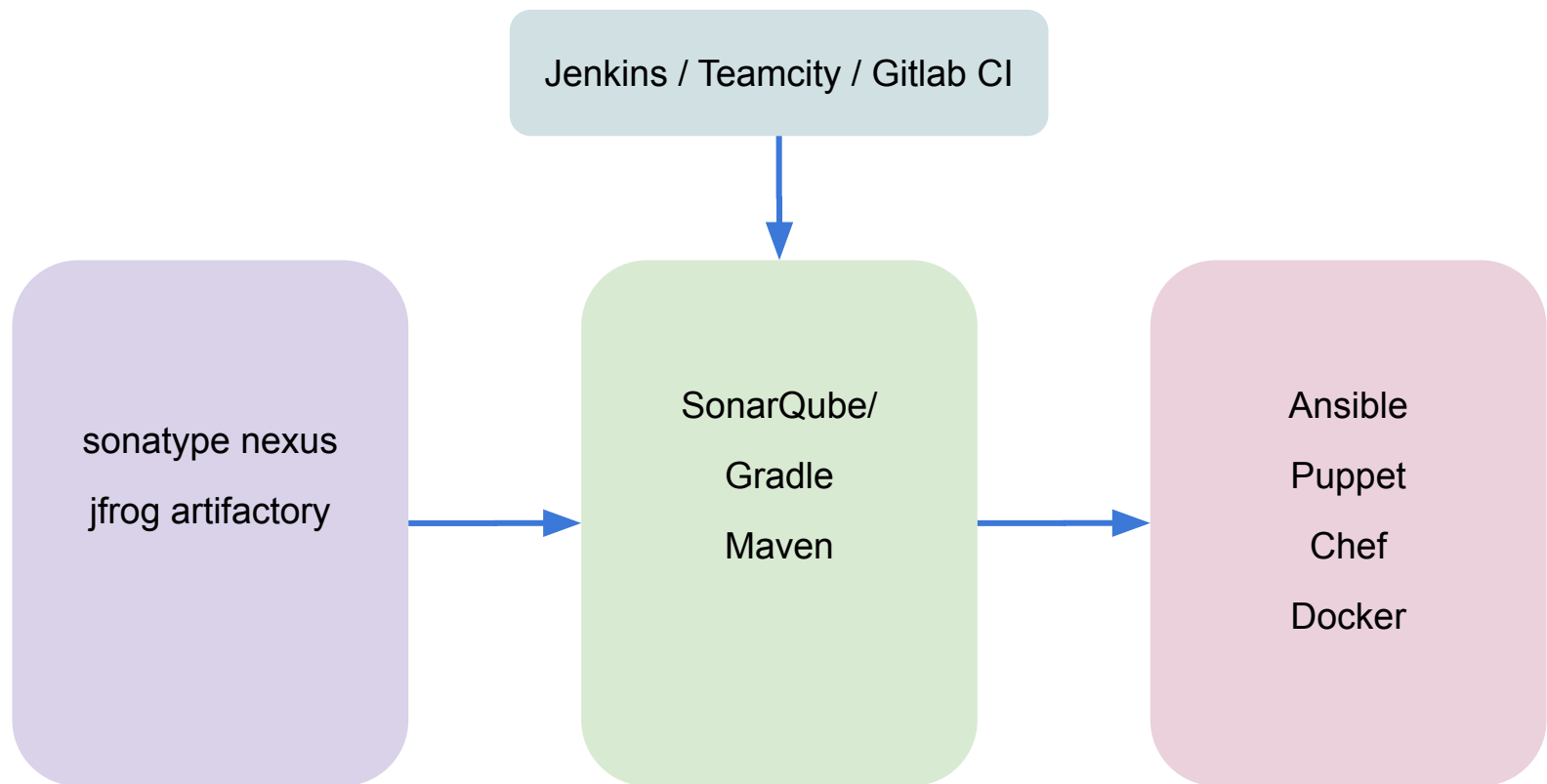
CD (Continuous Deployment, непрерывная установка) – этап непрерывной поставки продукта, который отвечает за **установку** дистрибутива продукта на продакшн окружение.

Иногда он **не** является этапом CI\CD, поэтому его появление опционально, но для полноценного конвейера и выполнения условий непрерывного цикла он необходим.

Continuous Deployment



Continuous Deployment



Continuous Deployment

Установкой может считаться:

- прогон плейбуков **Ansible**,
- запуск **Docker-контейнера** с образом.

Важно понимать, что администратор промышленной части, может, и будет проводить ручную проверку установленной версии продукта.

Continuous Deployment

Feedback – сбор обратной связи от клиентов (в том числе и внутренних) о продукте в виде **багов** и новых **story**.

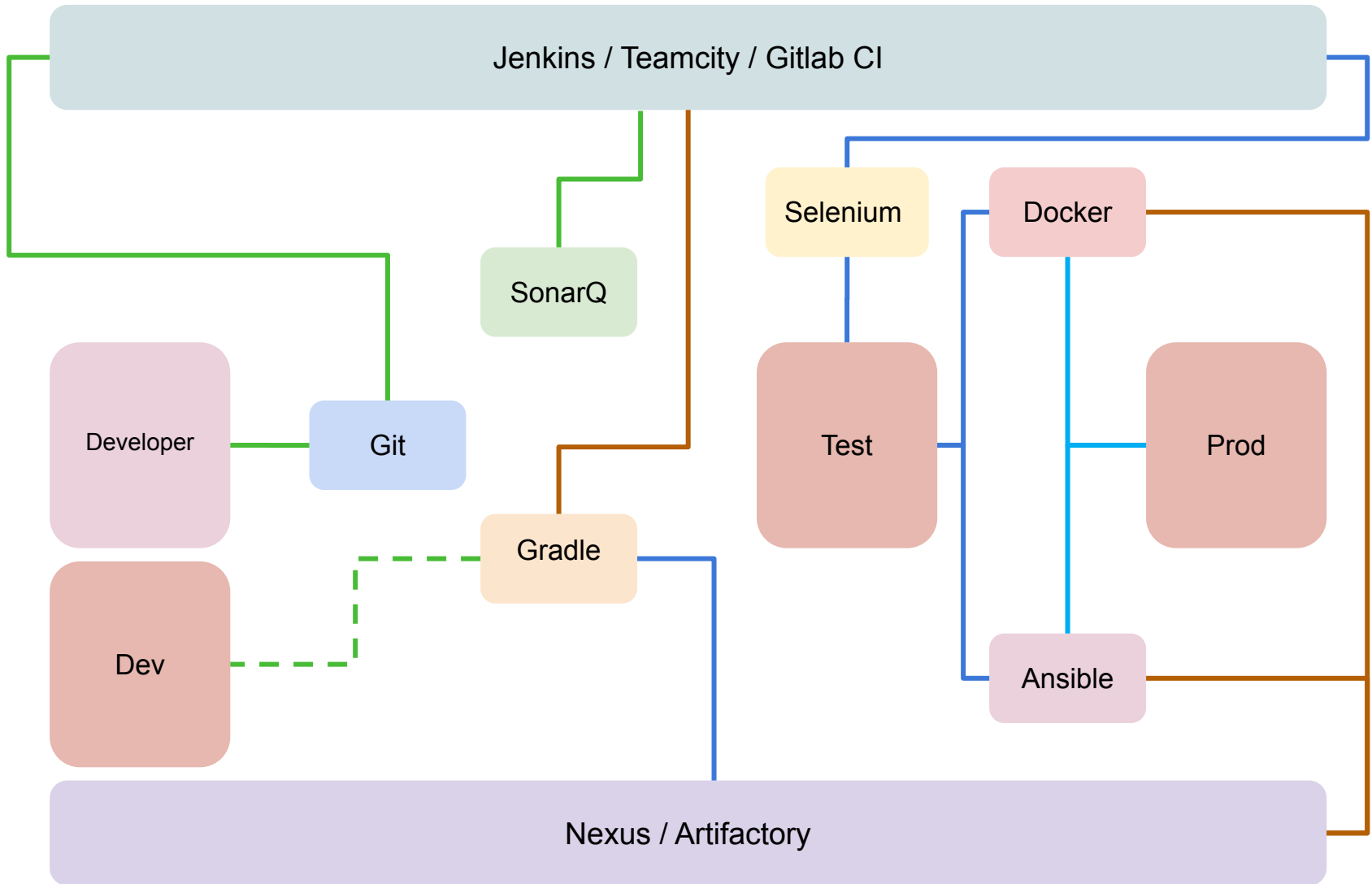
Важно помнить:

- ошибки являются нормой,
- даже на проде,
- ошибки являются ценностью,
- их необходимо правильно обработать,
- их необходимо правильно проанализировать,
- они требуют исправлений не меньше, чем внедрение новых фич.



Итоги

Итоговая схема



Итоги

- **CI\CD-процессы** отвечают за непрерывность конвейера производства программного продукта;
- На каждом из этапов в разных командах могут использовать **разный** набор инструментов;
- Набор наших задач, как специалистов может быть **разнообразный**:
 - поддержка инструментов **DevOps**,
 - поддержка взаимодействия (**автоматизация** конвейера),
 - поддержка отдельных этапов (**CI, CDL, CDP**),
 - внедрение и поддержка процессов **DevOps**,
 - **все** вышеперечисленные **задачи** сразу.

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Алексей Метляков