



Технология непрерывной поставки ПО

План

Занятие 1

Введение в CI/CD

Элементы CI/CD

Подготовка CI-сервиса

Занятие 2

CI/CD для контейнеризированного приложения

Описание окружений и деплоймовентов

Занятие 3

CI/CD поддерживающий процессы в организации

Построение CI/CD pipeline с учетом лучших практик

Практики DevOps

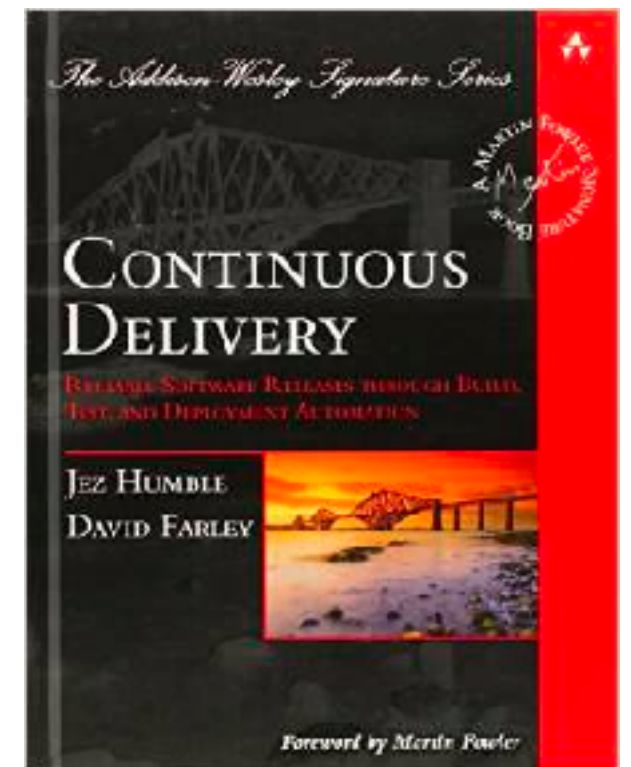
- Непрерывная интеграция (Continuous Integration)
- Непрерывная поставка (Continuous Delivery)
- Тестирование и Управление тестовыми данными
- Непрерывный мониторинг
- Управление версиями и релиз-менеджмент
- Управление конфигурацией и инфраструктура как код

Непрерывная интеграция

- проект забирается из СКВ
- проект собирается
- прогоняются тесты
- проходит выкатка на тестовый стенд (шоты)
- рассылаются оповещения

Непрерывная поставка

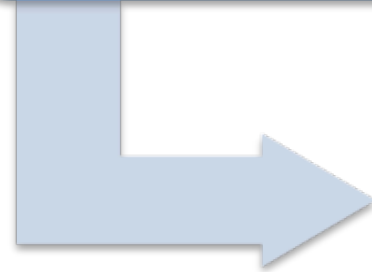
- Continuous Delivery (CD)
- Непрерывная поставка ПО — это процесс разработки и эксплуатации ПО, при котором каждое изменение может быть выкачено в боевую среду



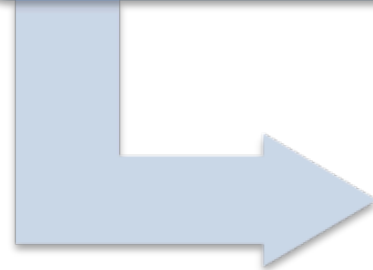
Конвейер



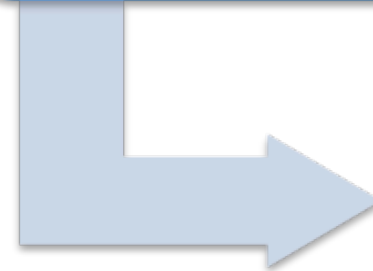
Среда
разработки



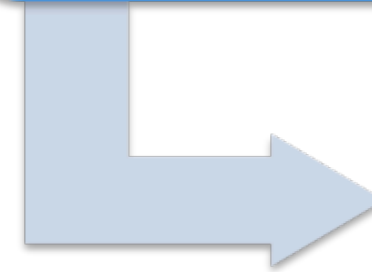
CI



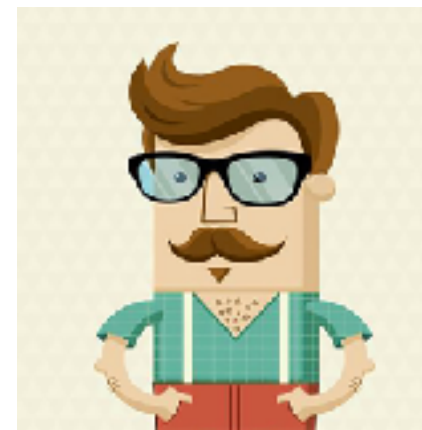
QA



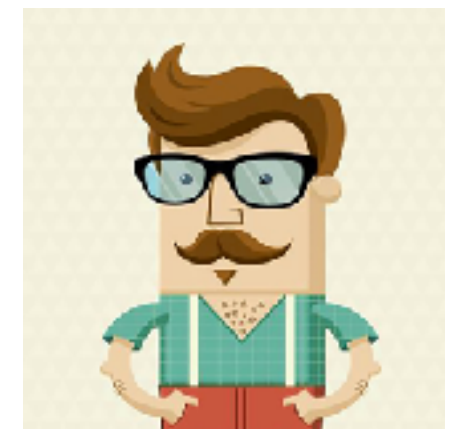
Staging



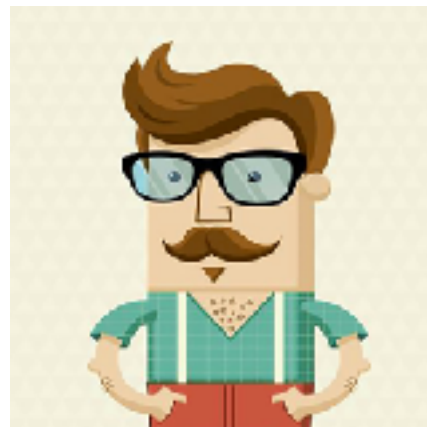
Production



Dev



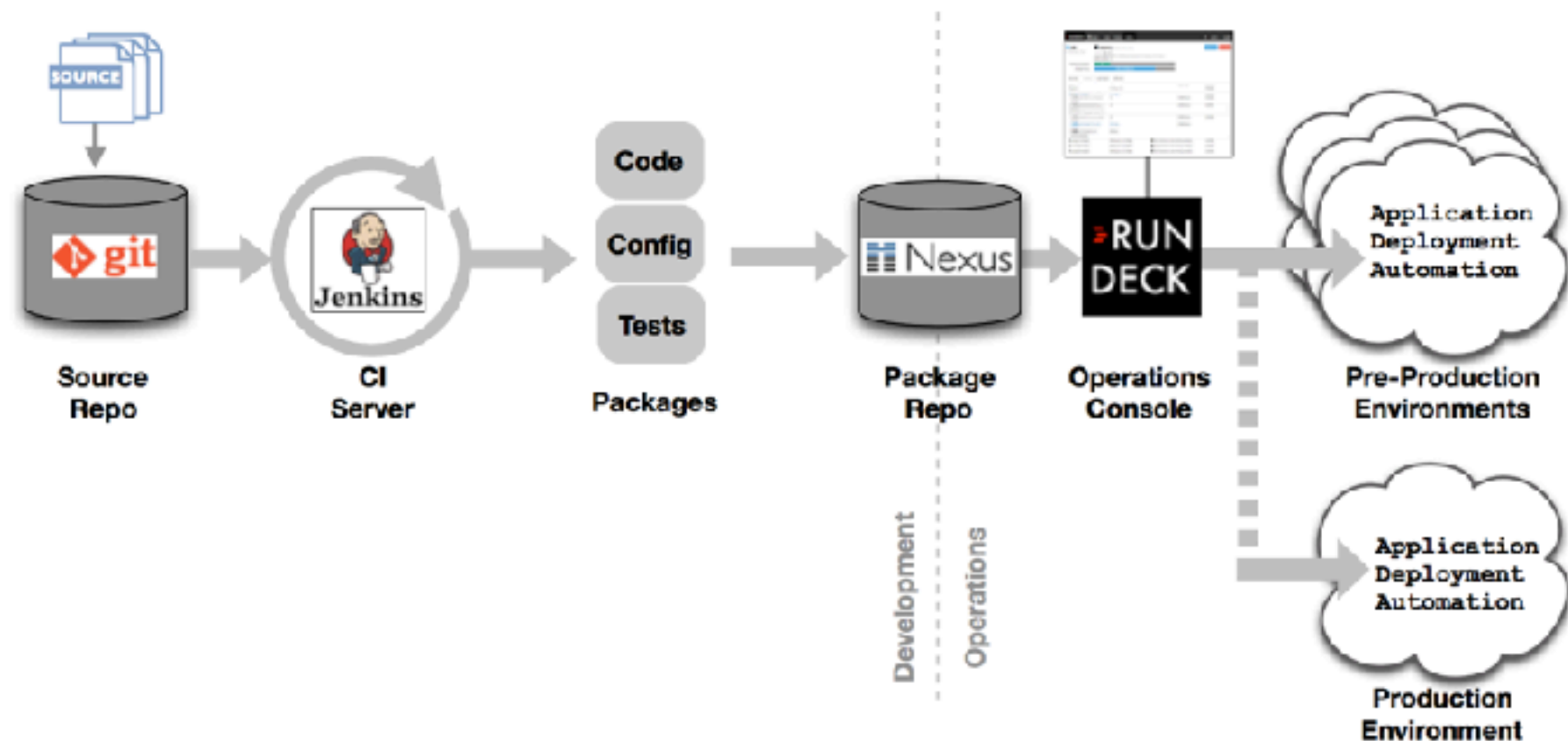
Ops



QA

CI/CD

- Автоматизация процессов (сборки, деплой, инфраструктура)
- Стандартизация подходов к управлению окружениями
- Повышение прозрачности процессов
- Обеспечение командного взаимодействия



Проблематика

- Разработка и Тестирование нуждаются в независимых окружениях.
- Окружений становится много, непонятно кто за них отвечает
- Состояние окружений не синхронизировано

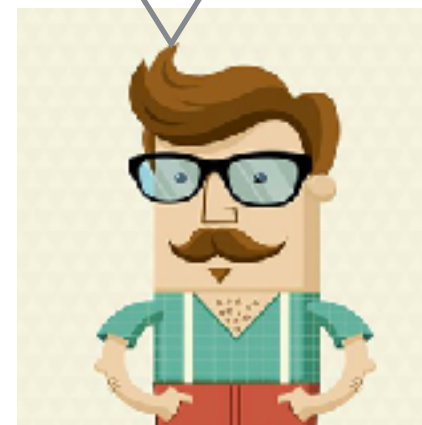
dev01.example.com

dev02.example.com

test01.example.com

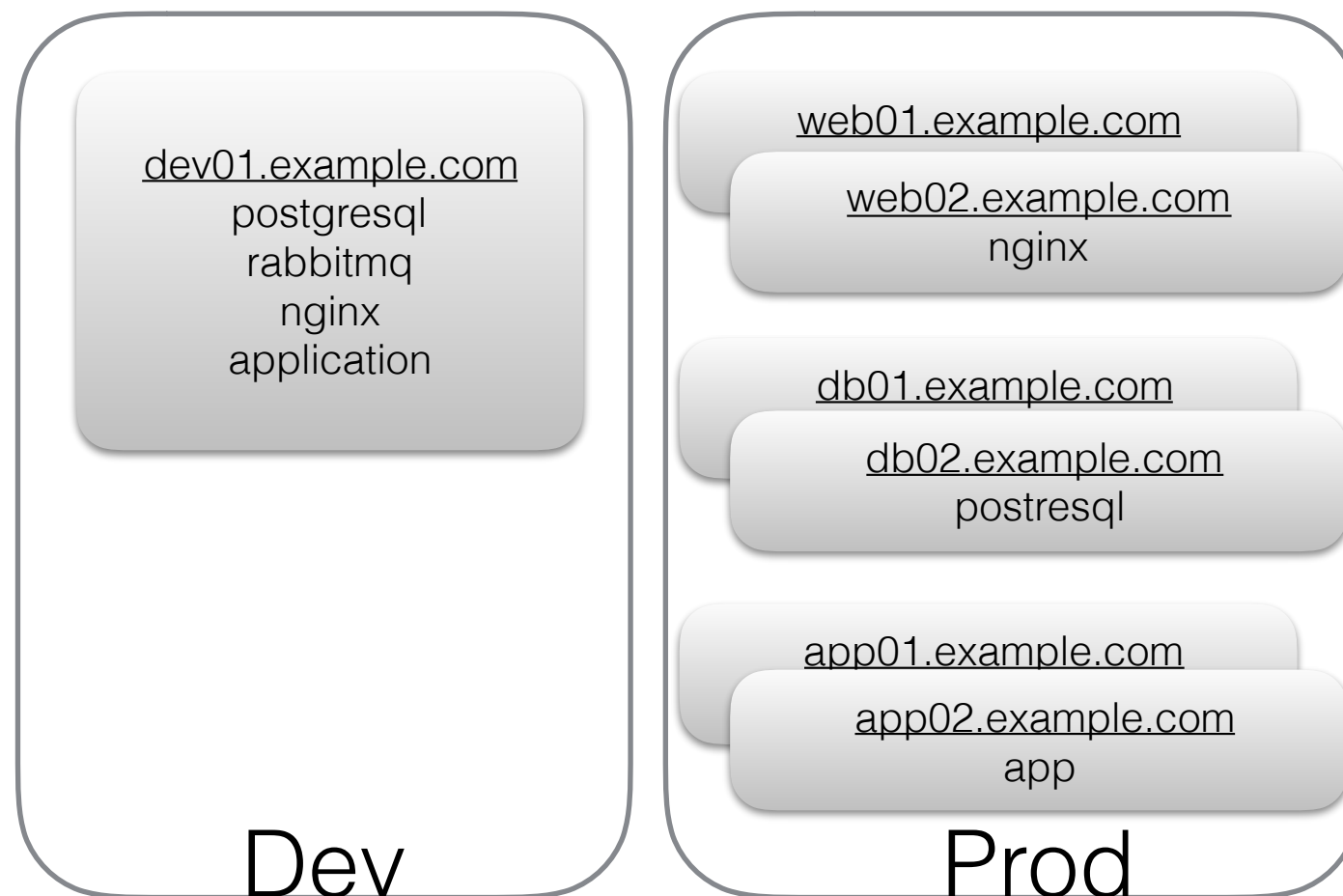
test02.example.com

На test01 развернута новая фича с изменениями в структуре таблиц БД и мы не можем там тестировать багфикс.
На dev02 кто-то обновил openssl, а на остальных – забыл.

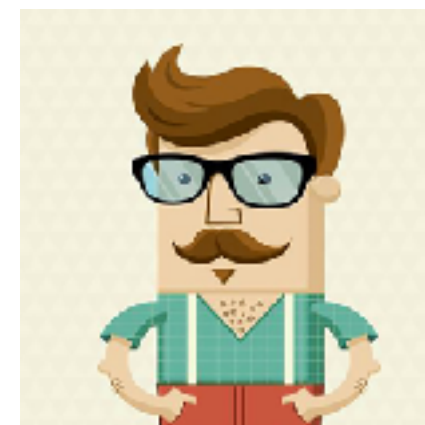


Проблематика

- Продукт может состоять из множества компонент
- Много компонент * много окружений = сложно
- Ограничения в ресурсах



Мы планируем запустить рекламную компанию и нужно провести нагрузочное тестирование на окружении аналогичном боевому



Процессы CI/CD

- Сборка приложения
- Контроль качества
- Публикация протестированных артефактов сборки в хранилище
- Создание нового окружения / Удаление более не нужного окружения
- Обновление окружения до состояния, описанного в виде кода
- Выкатка новой версии приложения / Откат до прошлой версии
- Применение изменений к базам данных этого окружения

Рассмотрим на примере

Веб приложение
Фронтенд: Angular2
Бекенд: Flask
База: PostgreSQL

web01.example.com
nginx, static files (html, js)

app01.example.com
python (api, flask app)

db01.example.com
postgresql

Please sign in

Email address
Password

Sign in

Пример 1: новый функционал приложения

Задача: добавить кнопку “Запомнить меня”

Фронтенд

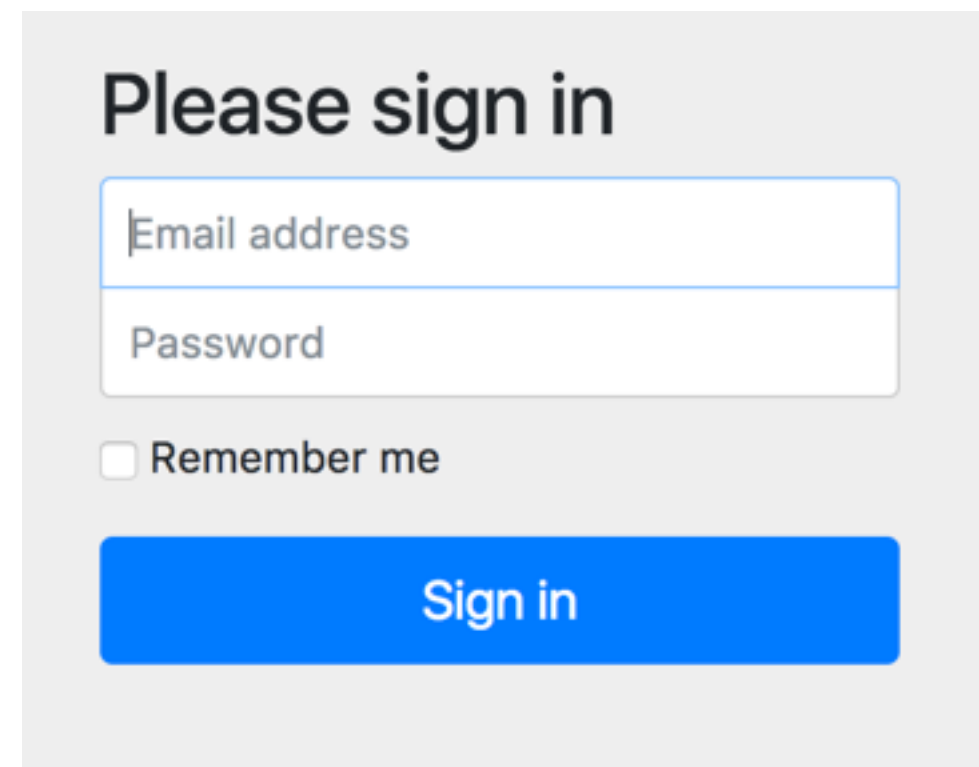
```
<form name="form" (ngSubmit)="f.form.valid && login()" #f="ngForm" novalidate>
  ...
  <div *ngIf="f.submitted && !password.valid" class="help-block">Password is required</div>
</div>
+ <div class="checkbox">
+   <label>
+     <input type="checkbox" value="{{remember_me}}"> Remember me
+
```

Бекенд

```
+@app.route("/session", methods=['POST'])
+def session():
```

БД

```
+CREATE TABLE session (
+  ...
+);
```



Пример 2: новая инфраструктура

Задача: добавление функционала рассылки

```
- hosts: email-servers
  roles:
    - role: postfix
      postfix_config_templates:
        -
          src: 'main.cf.j2'
          dest: '/etc/postfix/main.cf'
          owner: root
          group: root
          mode: '0644'
      postfix_tld: 'example.com'
```

dev

test

prod

Пример 3: Security disaster



Задача: Случился heartbleed. обновить openssl на всех серверах.

Как CI/CD помогает?

Контекст

Dev

Контекст'

Test

Контекст''

Prod

Изменение

2 docker-compose.yml

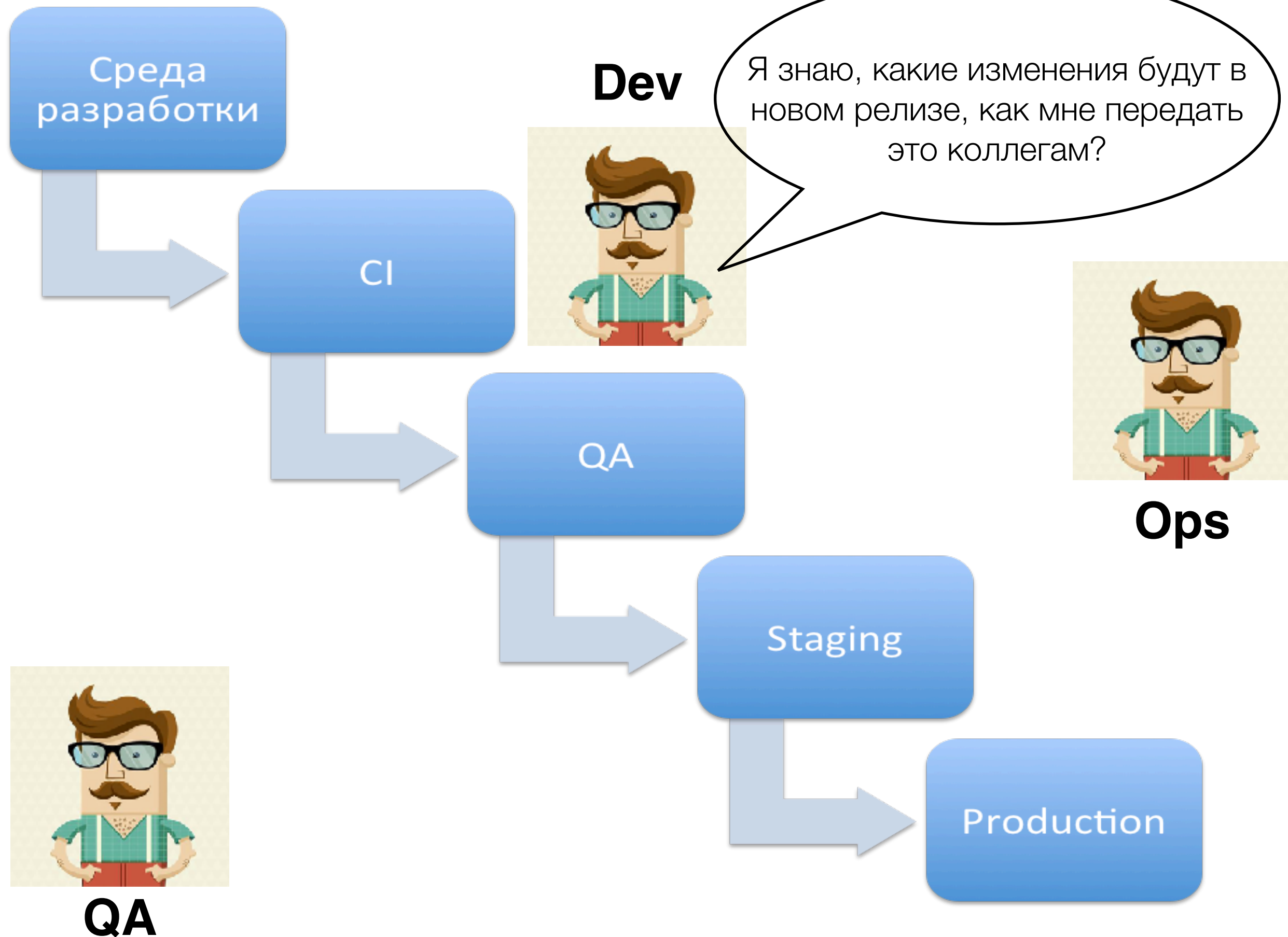
		@@ -8,4 +8,4 @@ services:
8	8	volumes:
9	9	- ./future-devops:/future-devops
10	10	- ./static:/static
11	-	- ./default.conf:/etc/nginx/conf.d/default.conf
11	+	- ./nginx.conf:/etc/nginx/nginx.conf

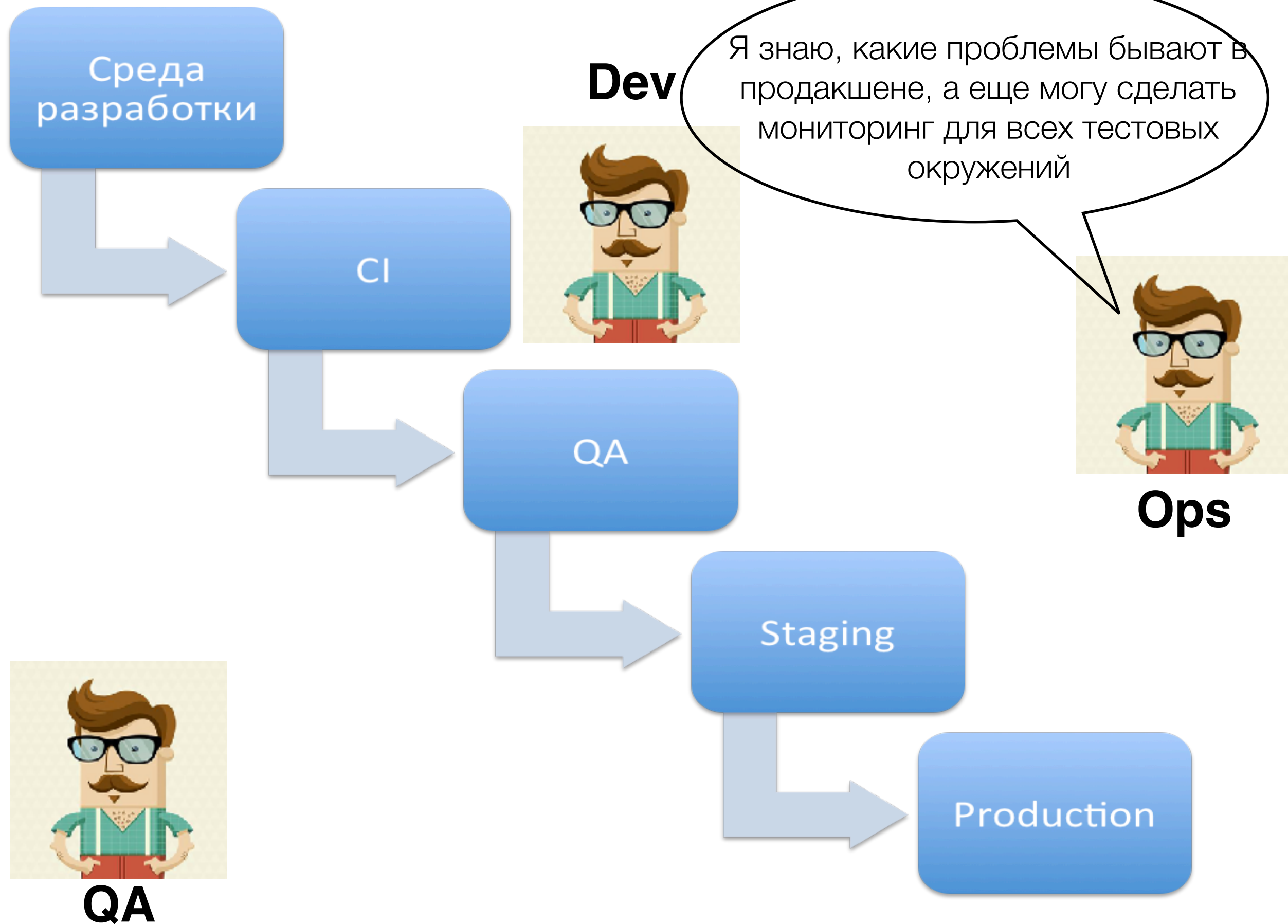
Принципы

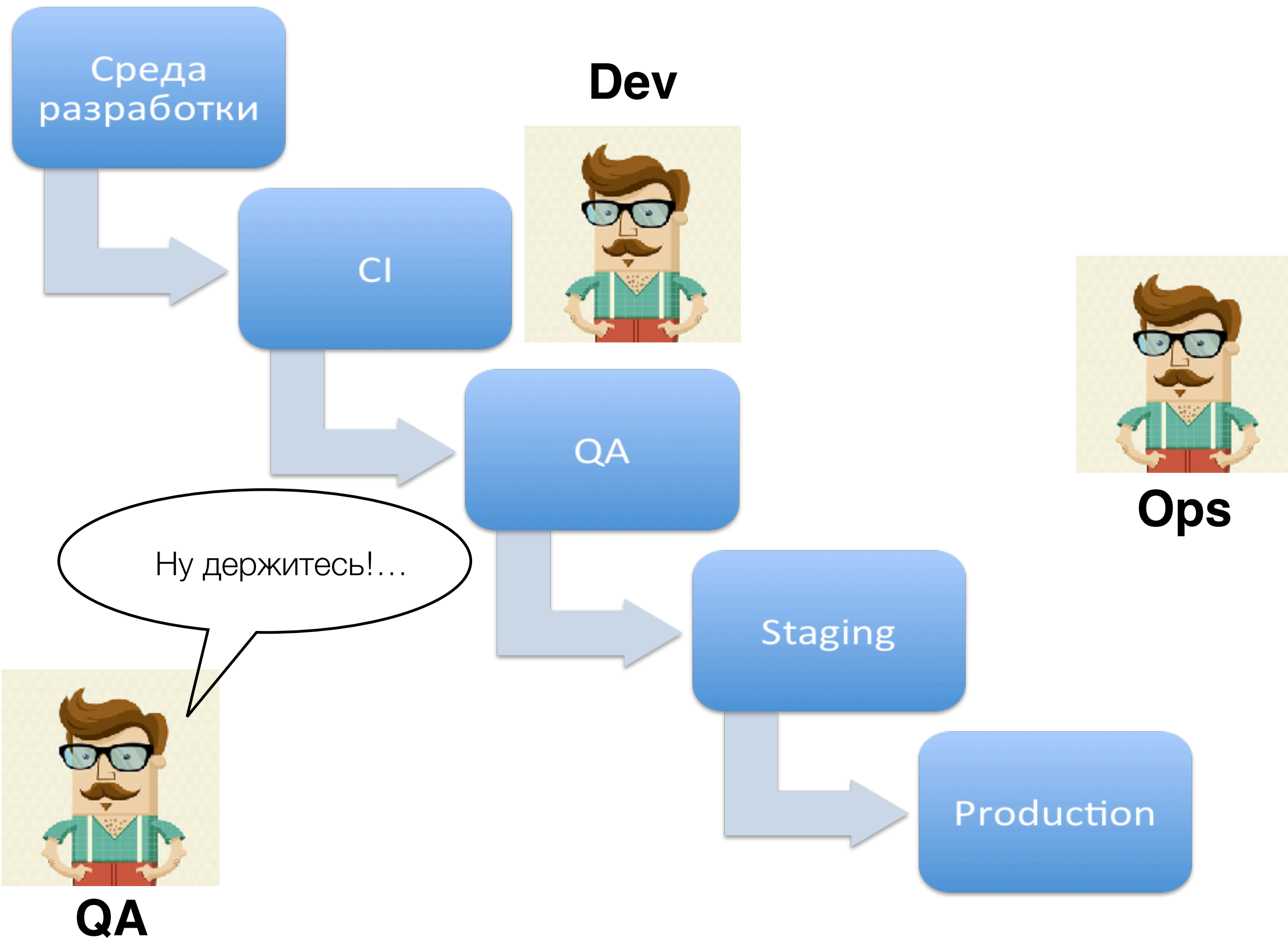
- Создать повторяемый и надежный процесс поставки ПО
- Автоматизировать все, что можно. Интерфейс с кнопками
- Хранить все в системе контроля версий
- Получать раннюю обратную связь
- Совершенствоваться через повторения
- Встроенный контроль качества
- Выполнено, значит зарелизилось
- Релизим именно то, что было протестировано
- Каждый отвечает за процесс поставки ПО

Принципы

- Создать повторяемый и надежный процесс поставки ПО
- Автоматизировать все, что можно
- Хранить все в системе контроля версий
- Получать раннюю обратную связь
- Совершенствоваться через повторения
- Встроенный контроль качества
- Выполнено, значит зарелизилось
- Релизим именно то, что было протестировано
- **Каждый отвечает за процесс поставки ПО**







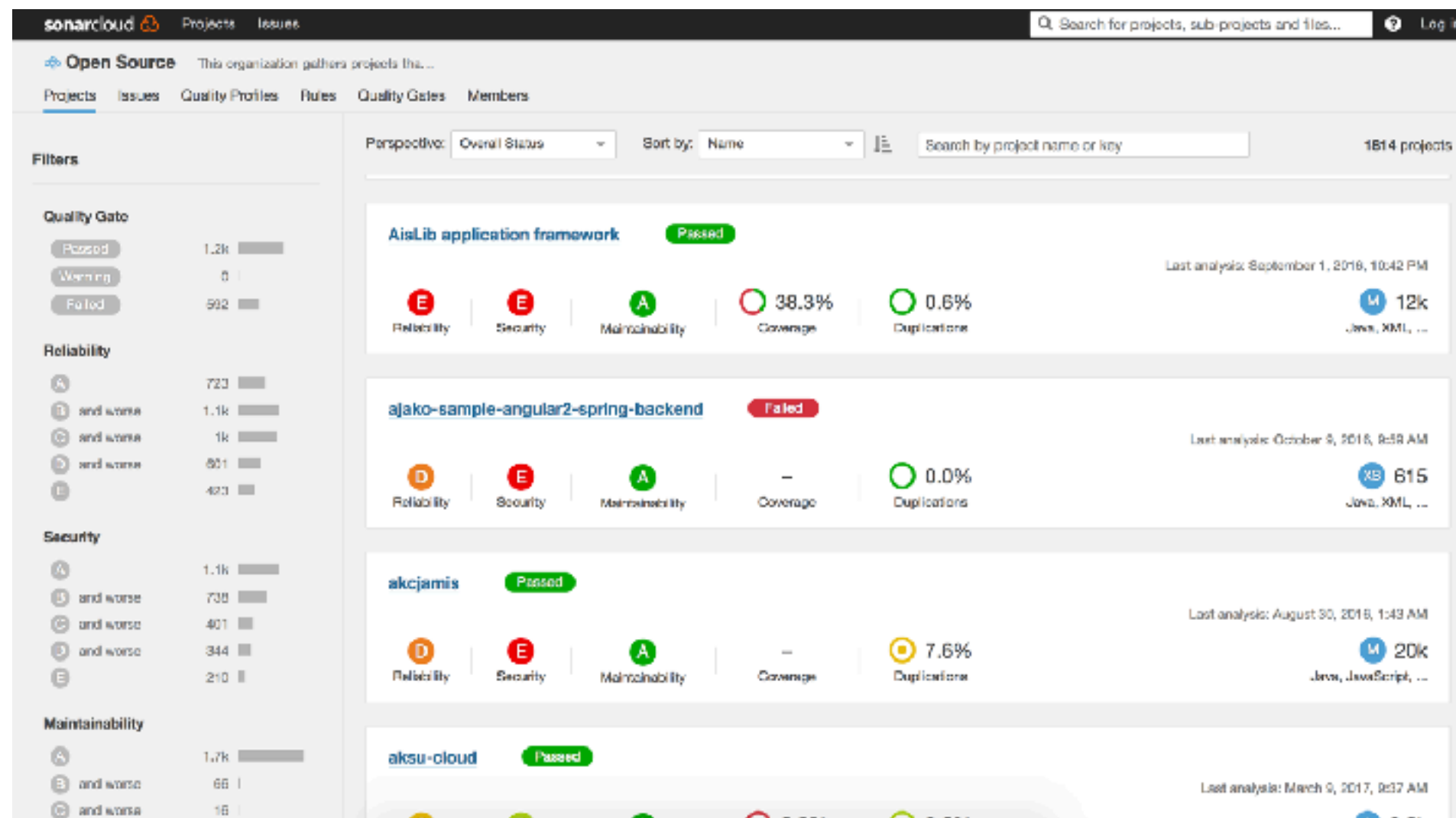
Обратная связь

- Светофор
- Отчеты о тестировании
- Контроль качества кода
- Уведомления о статусах в чат или почту
- Интеграция с task-треккером

Контроль качества кода

- Измерения и визуализации
 - Покрытие тестами
 - Технический долг
 - Поиск потенциальных багов и уязвимостей
- Помогает найти потенциальные проблемы
- Помогает отслеживать тренды

SonarQube



SonarQube

The screenshot displays the SonarQube web application interface. At the top, there's a navigation bar with 'sonarcloud' logo, 'Projects', and 'Issues' tabs. A search bar and a 'Log In' link are also present. Below the navigation bar, the breadcrumb trail shows 'Open Source / Activiti'. The main content area is divided into three sections:

- Left Sidebar:** Contains a filter for '5 / 342 Issues' and a list of issues. The first issue is 'Cast one of the operands of this multiplication operation to a "double".' with a 'Bug' icon and 'Minor' severity. The second issue is 'Equality tests should not be made with floating point values.' with a 'Bug' icon and 'Major' severity.
- Center Panel:** Displays the source code for the file 'Activiti - BPMN Layout / src/main/java/org/activiti/bpmn/BpmnAutoLayout.java'. The code is shown with line numbers (482 to 489) and a red vertical line indicating the current position. A red box highlights a specific issue in the code: 'NullPointerException might be thrown as 'closestPoint' is nullable here'.
- Right Panel:** Provides details for the highlighted issue. It shows the issue type as 'Bug', severity as 'Major', and a 'Reopened' status. It also indicates the issue was 'Not assigned' and has a '10min effort' estimate. The issue was created '2 years ago' by user '1409'.

Хранение артефактов

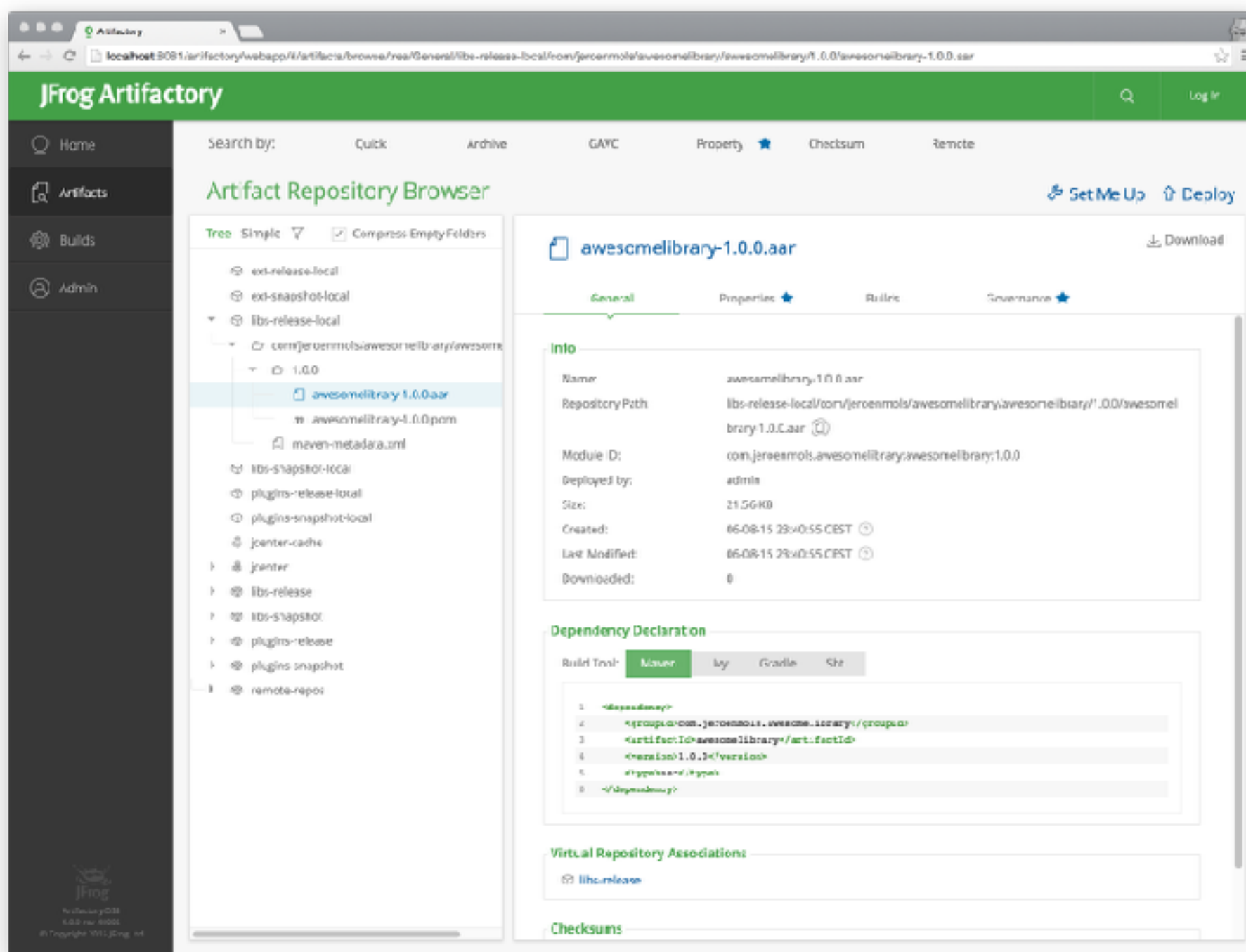
Результатом билда иногда является артефакт
Например: jar, war, rpm, zip, docker image

Исходный код	Артефакт
Текст	Бинарник
Diffable	Not diffable
Версионирование по содержимому	Версионирование по имени
Перезаписывается	Не должен перезаписываться

Хранение артефактов

- Artifactory
 - Nexus
 - Archiva
-
- Прокси для внешних репозиториев
 - Управление жизненным циклом артефакта
 - Модель управления доступом
 - Поиск
 - REST API
 - Single source of truth

Хранение артефактов

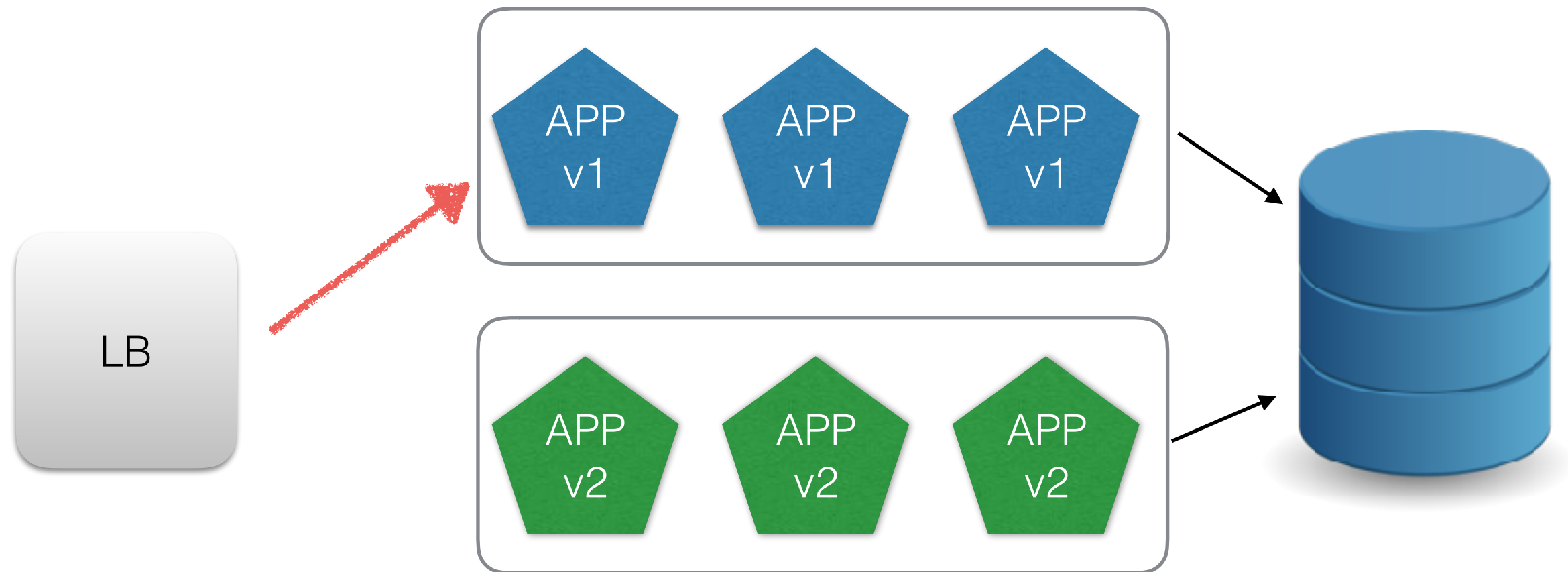


Стратегии выкатки

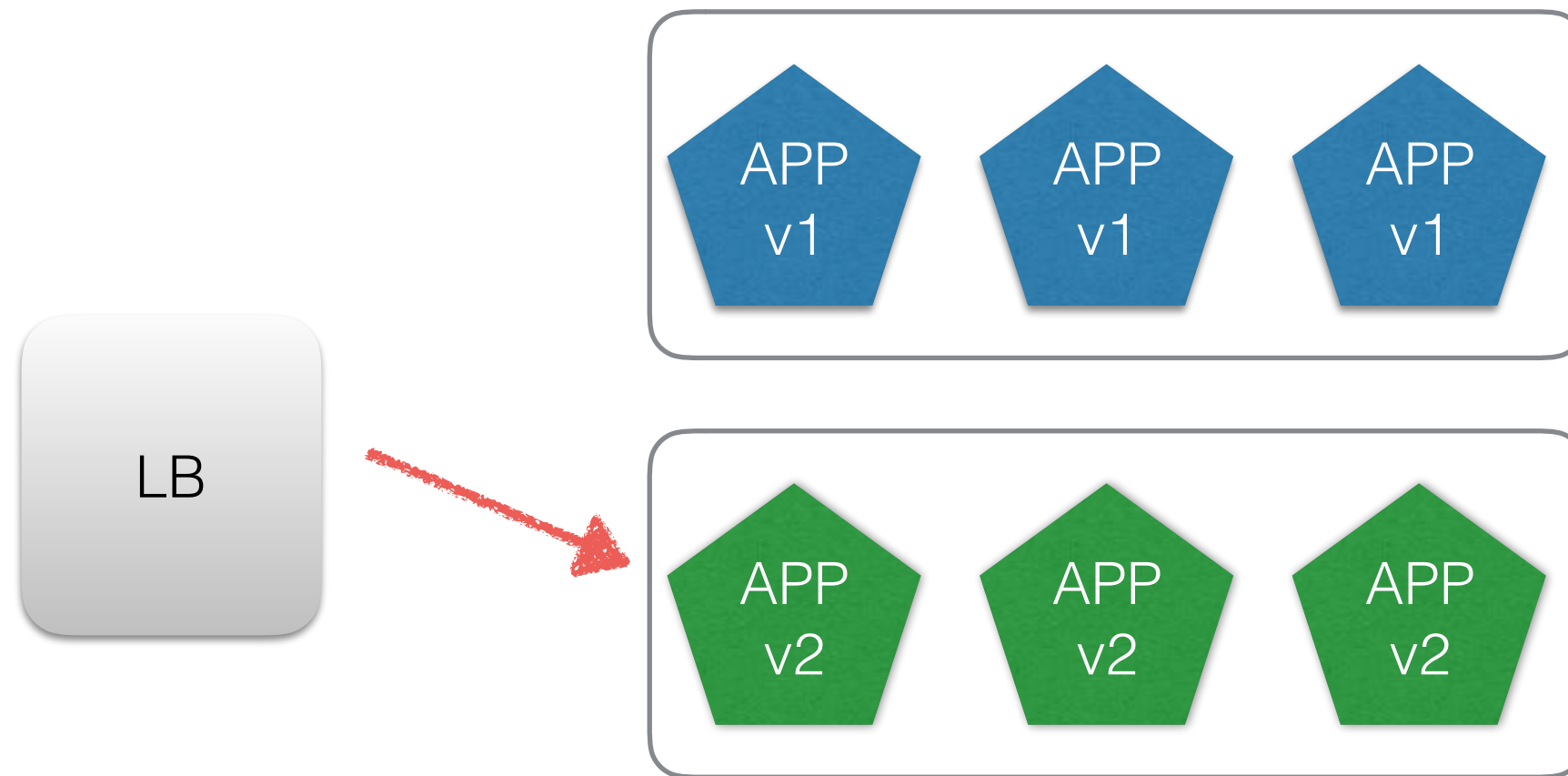
Для того, чтобы нам автоматически выкатывать изменения нужно реализовать обновление без простоев (zero-downtime deploy)

- Blue-green
- Canary
- Rolling

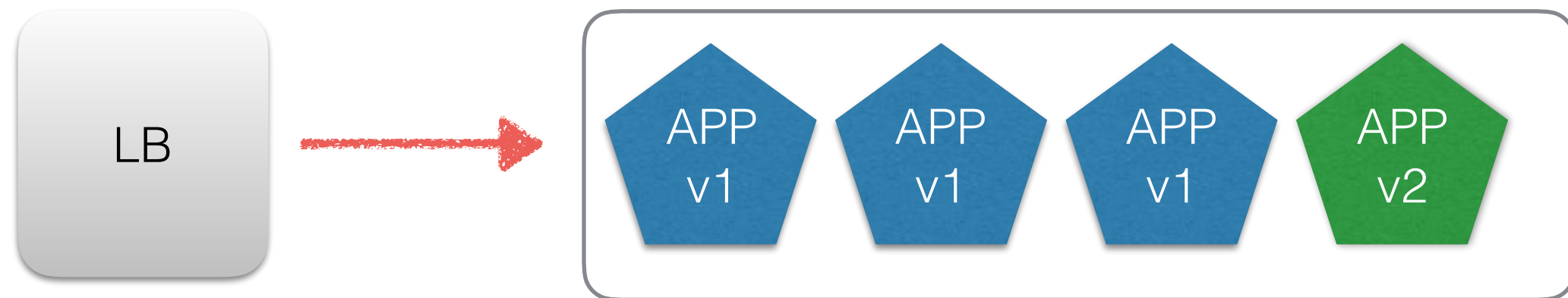
Blue-green deployment



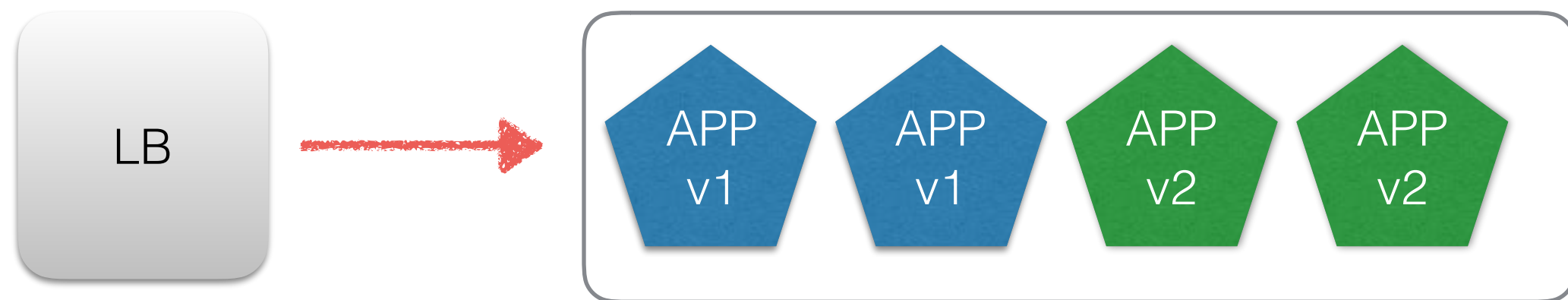
Blue-green deployment



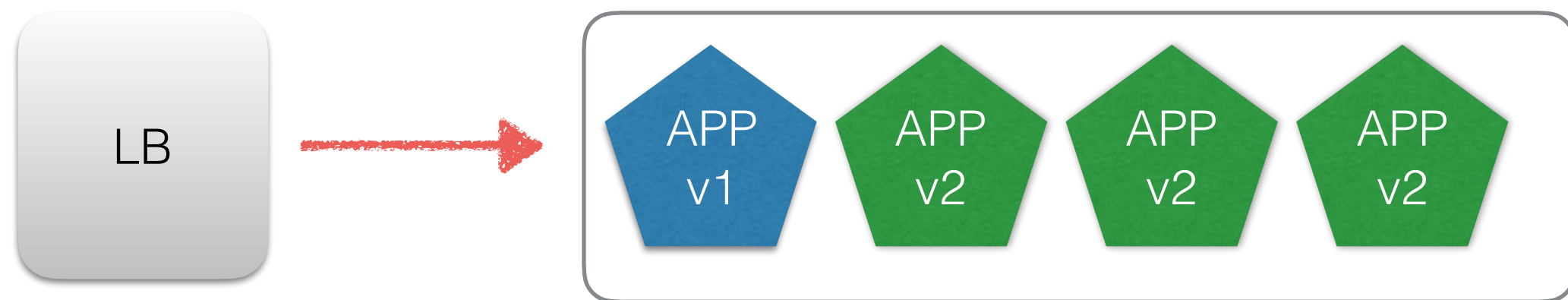
Canary deployment



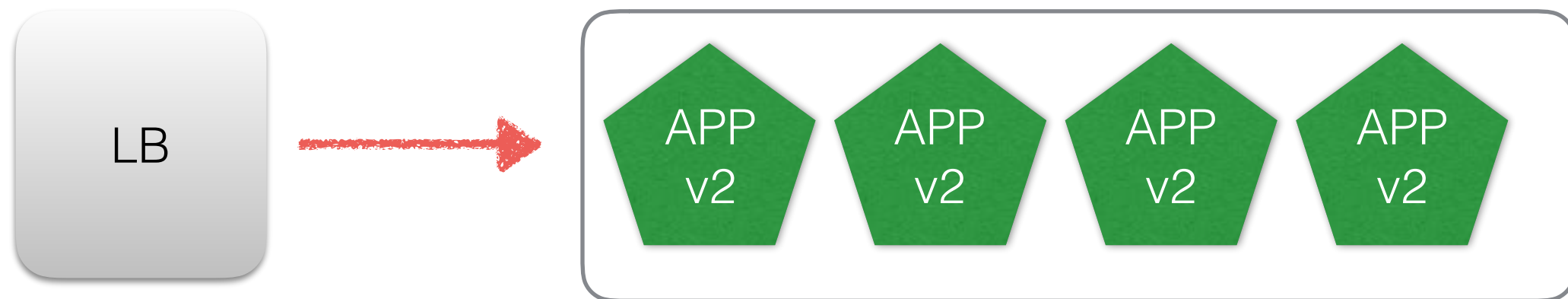
Rolling deployment



Rolling deployment



Rolling deployment



Нюансы

- Интеграция инструментов в процессы CI/CD
 - Как по созданию ветки в git создать окружение?
 - Как будет запускаться ansible, настраивающий окружения?
 - Как подключать новые окружения к мониторингу?
- CI-сервис в больших организациях
 - Windows / Linux / AIX / Docker
 - Много продуктов со своей спецификой
 - Состояние сборочных агентов уплывает
- CI/CD для компонент, поставляемых вендором

Метрики

- Скорость
- Качество

Метрики скорости

- Mean Lead Time – время на билд, тесты, деплой
- Daily Change Rate – успешных изменений в день
- Mean Time To Environment – время на подготовку нового окружения
- Mean Time To Detect – как долго баг может быть необнаруженным

Метрики качества

- Build Failure Rate – процент неуспешных сборок
- Deployment Failure Rate – процент неуспешных выкаток
- Infra-Related Failure Rate – процент неуспешных выкаток/сборок из-за проблем инфраструктуры
- Automated Detection Rate – процент проблем, выявленных на этапах тестирования

Инструменты

- Jenkins
- Bamboo от Atlassian
- TeamCity от JetBrains
- GitLab CI
- TravisCI
- Concourse CI
- Go CD
- Rundeck

Почему Gitlab CI

- Уже интегрирован в экосистему Gitlab (VCS, Issues, Docker registry!)
- Отдельный CI процесс под репозиторий (компонент)
- Быстрая обратная связь
- Визуализация процесса
- Достойная поддержка Docker
 - Тестирование внутри контейнеров
 - Тестирование с учетом других сервисов в контейнерах
- Обратная связь из коробки (много метрик, мониторинг)

Gitlab VCS

The screenshot shows the GitLab web interface for a project named 'example-shell-app'. The top navigation bar includes links for Projects, Groups, Activity, Milestones, and Snippets. The left sidebar contains a menu with options like Overview, Details, Activity, Cycle Analytics, Repository, Issues, Merge Requests, CI / CD, Wiki, Snippets, and Settings. The main content area displays the project details, including the repository name, a list of files (72 KB), and a table of commits. The current commit is 'add README' by 'Yury Ignatov', committed less than a minute ago, with a commit hash of 'f48b91cc'.

GitLab Projects Groups Activity Milestones Snippets

example-shell-app

Overview Details Activity Cycle Analytics

Repository Issues Merge Requests CI / CD Wiki Snippets Settings

example-shell-app

Star 0 Fork 0 SSH git@35.198.76.199:root/example

Files (72 KB) Commit (1) Branch (1) Tags (0) README Add Changelog Add License Add Contribution guide Set up CI

master example-shell-app /

add README Yury Ignatov committed less than a minute ago f48b91cc

Name	Last commit	Last Update
README.md	add README	less than a minute ago

Gitlab Issues

The screenshot shows the GitLab web interface for a project named 'example-js-app'. The top navigation bar includes links for Projects, Groups, Activity, Milestones, and Snippets. The left sidebar contains a menu with options: Overview, Repository, Issues (selected), Merge Requests, CI/CD, Wiki, Snippets, and Settings. The main content area displays an issue titled 'Add more tests' with the description 'Our application has very low code coverage. Someone has to deal with it.' The issue is marked as 'Open' and was created 'less than a minute ago' by user 'yignatov'. Below the description are reaction buttons (thumbs up, thumbs down, smiley) and a 'Create a merge request' button. A comment box is visible with a 'Write' tab and a 'Preview' tab. The right sidebar contains a 'Todo' section with a 'Mark done' button, and a list of issue attributes: Assignee (yignatov), Milestone (None), Time tracking (No estimate or time spent), Due date (Oct 14, 2017 - remove due date), Confidentiality (Not confidential), 1 participant (yignatov), Notifications (Unsubscribe), and Reference (root/test#1).

GitLab Projects Groups Activity Milestones Snippets

example-js-app

Overview Repository Issues 1 List Board Labels Milestones

Merge Requests 0 CI/CD Wiki Snippets Settings

Collapse sidebar

yignatov > example-js-app > Issues > #1

Open Opened less than a minute ago by yignatov Edit Close issue New issue

Add more tests

Our application has very low code coverage.
Someone has to deal with it.

0 0

Create a merge request

Write Preview B I

Write a comment or drag your files here...

Markdown and quick actions are supported Attach a file

Comment Close issue

Todo Mark done

Assignee yignatov @root Edit

Milestone None Edit

Time tracking No estimate or time spent

Due date Oct 14, 2017 - remove due date Edit

Confidentiality Not confidential Edit

1 participant yignatov

Notifications Unsubscribe

Reference: root/test#1

Gitlab Pipelines

The screenshot displays the GitLab web interface for a project. The top navigation bar includes links for Projects, Groups, Snippets, and Help, along with a search bar and a 'Sign in / Register' button. The left sidebar shows the project's overview, repository, issues (9,227), merge requests (481), and CI/CD section. The CI/CD section is expanded, showing 'Pipelines', 'Jobs', 'Schedules', and 'Charts'. The main content area shows the details of Pipeline #12682265, which is currently running. It was triggered 8 minutes ago by Grzegorz Bizon. The pipeline title is 'Extract ensure stage service from commit status class'. Below the title, it indicates 85 jobs from the commit `fix/gb/ensure-that-job-belongs-to-stage` (queued for 13 seconds). The commit hash `581a19f4` is also visible. The pipeline stages and jobs are as follows:

Build	Prepare	Test	Post-test
<ul style="list-style-type: none">build-packagereview-docs-d...	<ul style="list-style-type: none">retrieve-tests-...setup-test-env	<ul style="list-style-type: none">codequalitydb:check-sch...db:migrateres...db:migrateres...db:rollback-m...db:rollback-pg	<ul style="list-style-type: none">coverageflaky-example...lint:javascriptr...

Gitlab Pipelines

The screenshot displays the GitLab web interface. On the left is a sidebar with navigation links: Overview, Repository, Issues (9,228), Merge Requests (481), CI/CD (selected), Pipelines, Jobs, Schedules, Charts, Snippets, and Members. The main area shows the log for a job named 'setup-test-env'. The log text is as follows:

```
Showing last 500.3056640625 KiB of log - Complete Raw
Fixing public/assets/notify-abfe38022d82287585a29081cc27c46d0cf2f8bf68481bd421ba1ce23ca7c598.css
Fixing public/assets/notify-abfe38022d82287585a29081cc27c46d0cf2f8bf68481bd421ba1ce23ca7c598.css.gz
$ bundle exec rspec -i spec -e 'require "spec_helper" ; TestEnv.init'
Knapsack time offset warning enabled!

==> Setting up GitLab Shell...
Warning
You are running as user root, we hope you know what you are doing.
Things may work/fail for the wrong reasons.
For correct results you should run this as user git.

mkdir -p /builds/gitlab-org/gitlab-ce/tmp/tests/.ssh: OK
chmod 700 /builds/gitlab-org/gitlab-ce/tmp/tests/.ssh: OK
mkdir -p /builds/gitlab-org/gitlab-ce/tmp/tests/repositories: OK
chmod u+rwx,o-rwx /builds/gitlab-org/gitlab-ce/tmp/tests/repositories: OK
mkdir -p /builds/gitlab-org/gitlab-ce/tmp/tests/non-existent-repositories: OK
chmod u+rwx,o-rwx /builds/gitlab-org/gitlab-ce/tmp/tests/non-existent-repositories: OK
Creating/Repairing hooks symlinks for all repositories
done
GitLab Shell setup in 40.522782655 seconds...

==> Setting up GitLab...
Warning
You are running as user root, we hope you know what you are doing.
Things may work/fail for the wrong reasons.
For correct results you should run this as user git.

GitLab setup in 28.153000021 seconds...
$ scripts/gitlab-test-build
if test -z "/usr/local/bundle" ; then cd ruby && bundle config --local path vendor/bundle; fi
cd ruby && bundle install
Warning: the running version of Bundler (1.14.6) is older than the version that created the lockfile (1.15.4). We suggest you upgrade to the latest version of Bundler by running `gem install bundler`.
Fetching gem metadata from https://rubygems.org/.....
Fetching version metadata from https://rubygems.org/.

***
```

On the right, the job details for 'setup-test-env' are shown:

- Merge Request: !14724
- Duration: 9 minutes 37 seconds
- Runner: #37398
- Tags: [gitlab-org](#)
- Commit: [501a19f4](#) in !14724
- Extract ensure stage service from commit status class
- Pipeline #12682285 from [fix/gb/ensure-that-job-belongs-to-stage](#)
- Dropdown menu: prepare
- Job status: [→](#) [🔄](#) setup-test-env
- Job status: [🔄](#) retrieve-tests-metadata

Спасибо!

- Ждем ваших вопросов
- О готовности домашнего задания оповестим в Slack по готовности