

Что такое DevOps

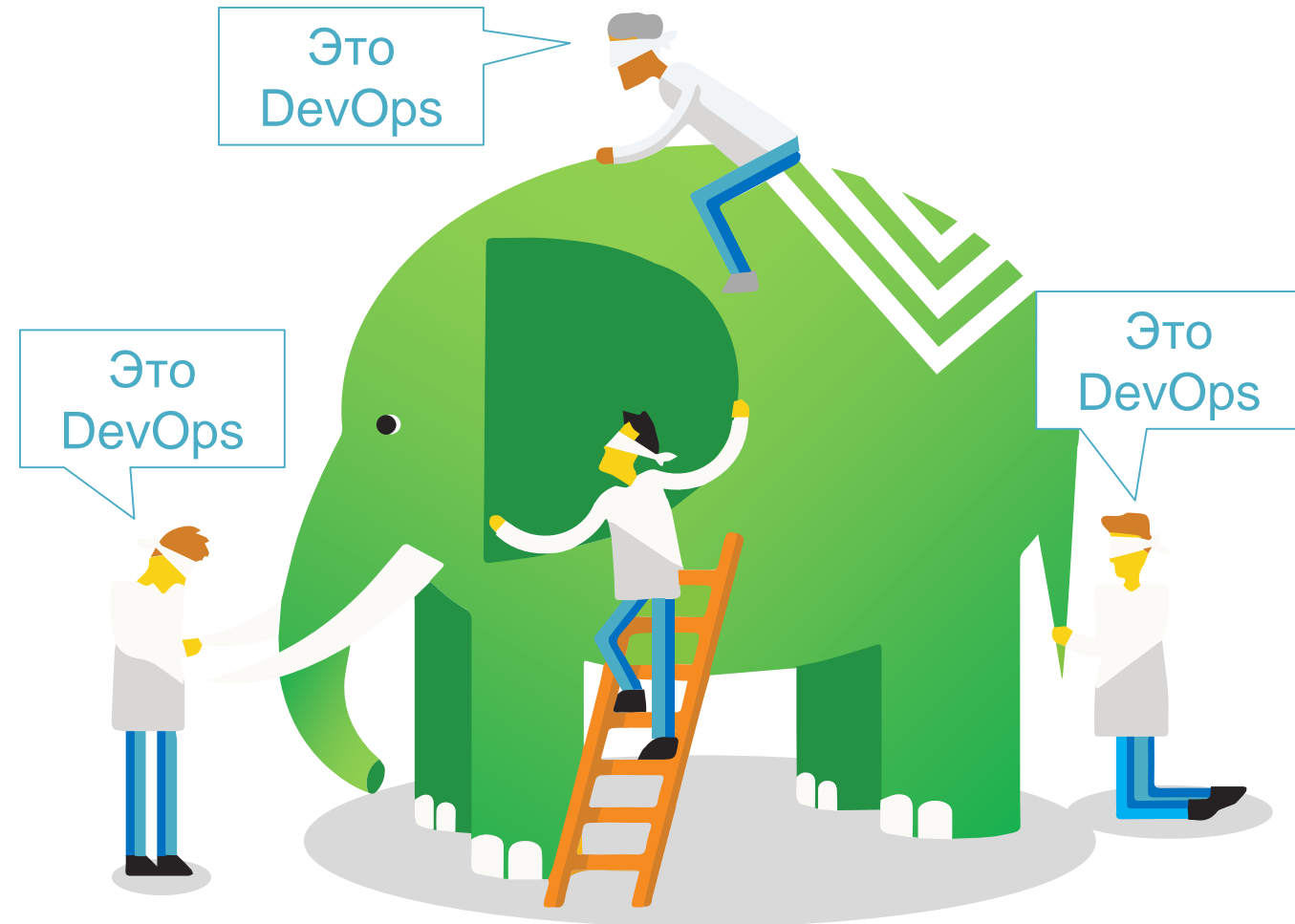
DevOps* – это инженерная культура, практики и инструменты, направленные на сокращение релизного цикла, повышение эффективности и обеспечение возможности выпуска релиза в любой момент посредством:

- интеграции IT Development и IT Operations;
- создания единой команды;
- автоматизации рутинных операций;
- частой проверки кода



ЦЕЛЬ: сократить T2M* и повысить гибкость внедрений (release anytime) без потери качества и надежности

Введение > CI > Среды > CD > CT > Метрики



DevOps – акроним от англ. development и operations

T2M (time-to-market) – срок от начала разработки продукта до его выхода на рынок

Заблуждения вокруг и около DevOps

Введение > CI > Среды > CD > СТ > Метрики

Некто из ИТ: А что, DevOps будет вместо Agile?

Некто из ИТ: DevOps и Waterfall несовместимы!

Менеджер ИТ: Мы в эти выходные не работаем, у вас будет время прийти и установить мне DevOps.

Dev: DevOps – это задача администраторов.

Dev: У меня установка на среды разработки идет по одной кнопке.
Я сделал DevOps!

Dev: Мне DevOps не нужен. Мы все гении разработки, а Вася делает сборку даже во сне.

Вопрос: Что будет, если количество задач увеличится, в вашу команду добавятся новые разработчики, а Вася уйдет в декрет?

Dev: Это проблема новых разработчиков, а Вася ...



Ops: Вы разработайте процесс развертывания и покажите, а мы выставим свои замечания.

Ops: У нас и так все хорошо и надежно. Не нужны нам новые инструменты.

Вопрос: Что будет, если не один релиз в месяц, а два в неделю?

Ops: Срочно увеличиваем штат!

Ops: Ну хорошо, внедрили DevOps. Где моя одна кнопка?

Менеджер ИТ: Надо плотно поработать три месяца, сделаем DevOps и расслабимся

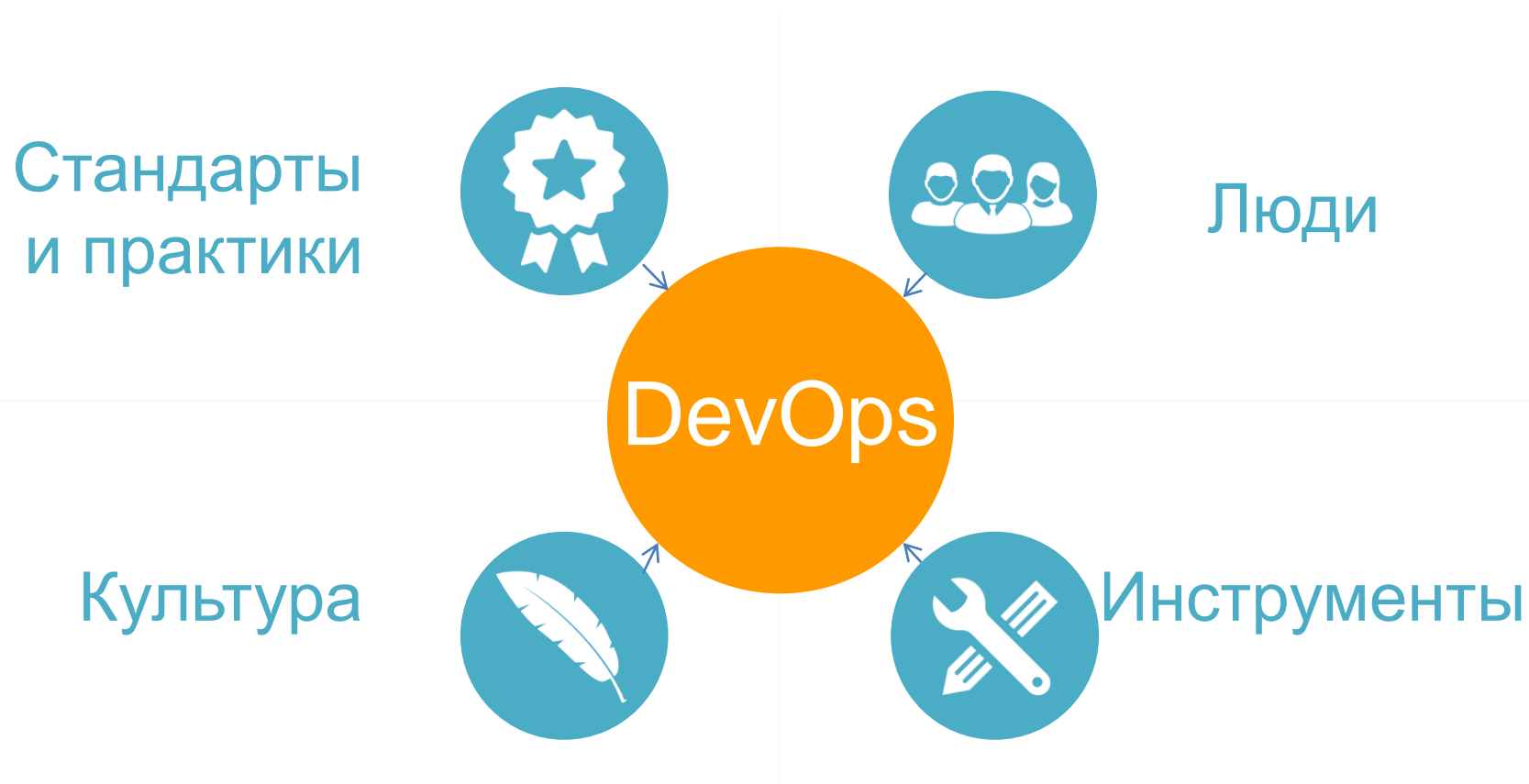
Вопрос: Вас не смущает слово *Continuous* в названиях практик DevOps?

Ops: Э-э-э ...



Основные компоненты DevOps

Введение > CI > Среды > CD > CT > Метрики



Только изменив **все четыре** составляющих, можно внедрить DevOps!

DevOps принципы

[Введение](#) > [CI](#) > [Среды](#) > [CD](#) > [СТ](#) > [Метрики](#)

Culture

- Развитие инженерной культуры – от разработчика к инженеру
- От ответственности за этап – к ответственности за результат

Automation

- Релиз в любое время
- Принятие изменений и уменьшение рисков

Measurement

- Непрерывное совершенствование основанное, на метриках и реальных данных

Sharing

- Коллаборация и открытая коммуникация
- Постоянная обратная связь «справа налево»

Основные практики DevOps, связанные в циклический автоматизированный процесс

Введение > CI > Среды > CD > СТ > Метрики

Непрерывное развертывание (Continuous Deployment)

Автоматическое развёртывание на средах вплоть до промышленной

Непрерывная поставка (Continuous Delivery)

Автоматическое развертывание и тестирование

Непрерывная интеграция (CI*)

Автоматическая сборка, запуск модульных тестов и статического анализа



- **Повышение качества кода за счет:**

- о практик модульного тестирования (unit testing)
- о инспекции кода (code review) и статического анализа кода на соответствие лучшим мировым стандартам кодирования и безопасности

- **Сокращение сроков разработки за счет:**

- о ускорения цикла обратной связи
- о увеличения частоты сборок (минимум 1 раз в день)

- **Повышение качества тестирования за счет:**

- о внедрения практик автоматизированного тестирования - testing automation
- о внедрения динамического анализа приложений на уязвимость

- **Сокращение времени вывода в ПРОМ за счет:**

- о автоматизации развертывания дистрибутивов на среды
- о динамического выделения инфраструктуры, infrastructure as code

- **Повышение частоты вывода релизов в ПРОМ за счет:**

- о внедрения практик автоматизации развертывания

*Continuous Integration

Практики автоматизированного тестирования

Continuous Testing

Введение > CI > Среды > CD > CT > Метрики



Виды автоматизированного тестирования

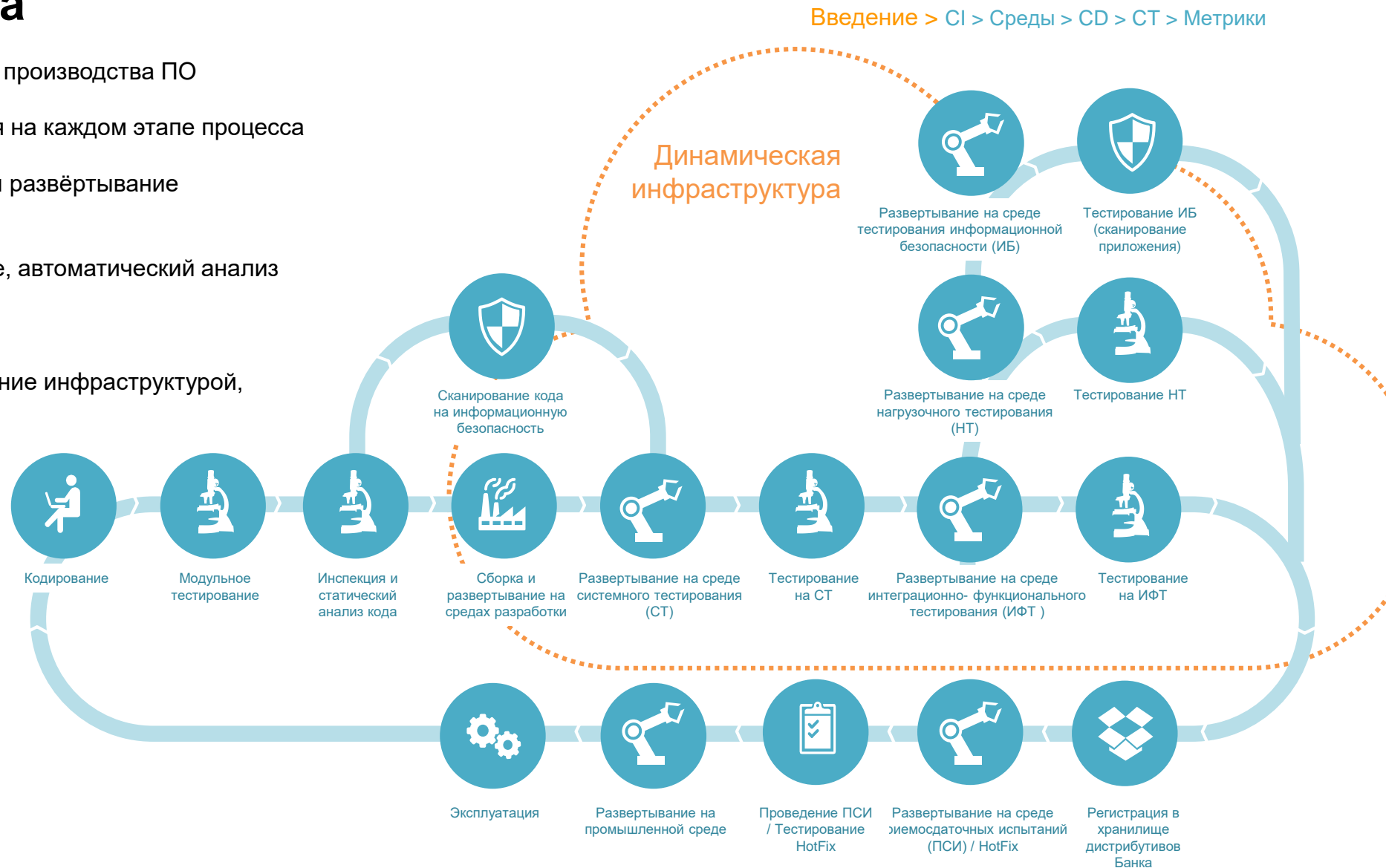
Что это такое?

Кто разрабатывает и проводит тесты?

Модульное тестирование – unit testing	Проверка работоспособности части кода (функции, класса и т.д.). Тесты пишутся на языке программирования с использованием специальных фреймворков	Инженер Agile - команды
Статический анализ кода и ИБ	Проверка кода на соответствие международным стандартам языков программирования и написания безопасного кода с помощью специальных инструментов.	
Системное тестирование	Проверка работоспособности изолированной части функционала системы (без интеграции).	
Интеграционно-функциональное тестирование	Проверка работоспособности сквозных бизнес-операций между разными системами для нового и для ранее разработанного функционала.	Инженер тестирования ДК
Нагрузочное тестирование	Проверка работоспособности системы при одновременной работе большого количества пользователей. Выполняется только для высоконагруженных систем и бизнес-операций	
Динамическое тестирование приложения на безопасность	Проверка системы на уязвимости (повышение привилегий, выполнение произвольного кода, sql injection,...) и «закладки»	

Циклический автоматизированный процесс на средах Банка

- Бесшовный сквозной процесс производства ПО
- Максимальная автоматизация на каждом этапе процесса
- Автоматизированная сборка и развёртывание дистрибутива
- Автоматическое тестирование, автоматический анализ качества кода, проверки кода и приложений на ИБ
- Автоматизированное управление инфраструктурой, инфраструктура как код, динамическое выделение инфраструктуры из облака

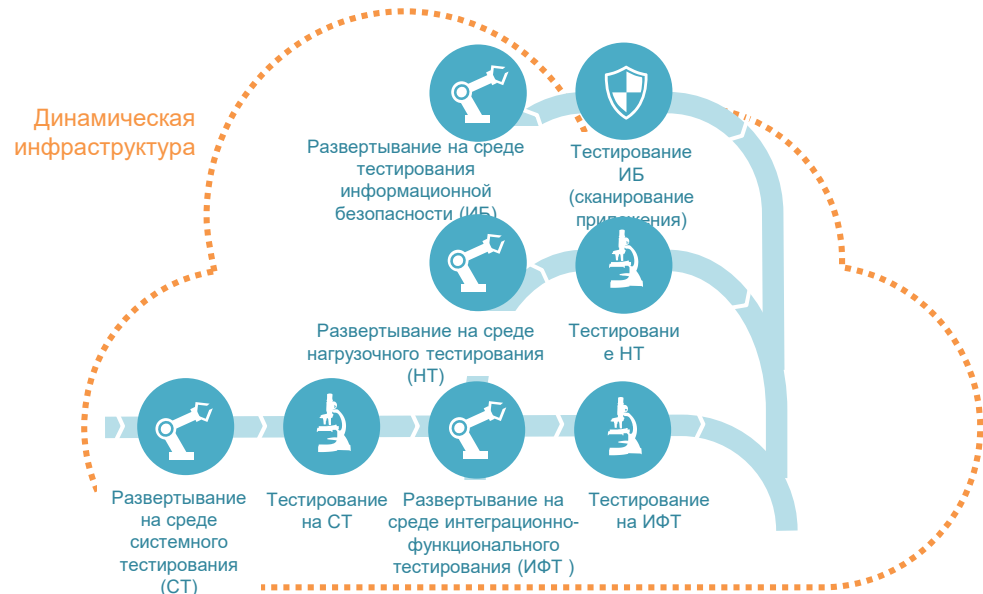


Основные практики DevOps

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.



Непрерывная поставка (CD_L, Continuous delivery) – практика разработки программного обеспечения, гарантирующая то, что программное обеспечение постоянно готово к развертыванию в промышленную эксплуатацию. Практика включает в себя автоматизацию развертывания на тестовые среды, автоматизацию тестирования успешности развертывания на уровне администраторов (технические и функциональные тесты развертывания), а также интеграцию с процессами автоматизированного функционального, нагрузочного тестирования и динамического тестирования на безопасность.



Непрерывное развертывание (CD_P, Continuous deployment) – практика разработки программного обеспечения, направленная на полную автоматизацию поставки от среды разработки в промышленную среду. Является дальнейшим развитием процесса Непрерывной поставки. В условиях Банка дополнительно затрагивает процесс регистрации в хранилище дистрибутивов Банка и автоматизацию приёмо-сдаточных испытаний. Характеризуется тесным взаимодействием разработки и эксплуатации.

Решаемые проблемы



Существующие проблемы производства

- 1 Из-за отсутствия общих стандартов ведения разработки (паттерны программирования, нотации и т.д.) и централизованного хранения кода случаются конфликты версий, пишется неоптимальный код, снижается эффективность разработки.
- 2 Разработка не выполняет интеграционные тесты, поэтому ошибки интеграции выявляются на поздних этапах – вплоть до ПСИ.
- 3 Ручное управление средами и развертыванием приводит к длительному ожиданию стендов командами (до нескольких месяцев) и увеличивает число ошибок при настройке стендов.
- 4 Большой объем ручного тестирования, отсутствие модульных и быстрых "smoke" тестов увеличивает время тестирования или сокращает объем тестового покрытия.
- 5 Проверка качества и безопасности кода при сборке не является обязательной. Поэтому в итоге можем получить небезопасный (с уязвимостями) и неоптимальный код (увеличение нагрузки на сопровождение).

Решение с помощью DevOps

 VC	Код хранится централизованно, любое изменение сохраняется с новой версией. Разработка ведется согласно стандартам, адаптированным к команде.
 AT	При каждой сборке выполняются быстрые smoke-тесты, проверяющие интеграцию "на заглушках" и эмуляторах, ошибки выявляются сразу при сборке.
 AD Iaas DI	Выделение среды происходит за минуты из облака Банка. Развертывание и конфигурирование автоматизировано.
 AT	Весь код покрыт модульными тестами, тестирование максимально автоматизировано.
 ACR PrCR IS JaaS	Код и готовое приложение проверяются на уязвимости и на качество – как в Apple и Google все события централизованно журналируются.

На рынке присутствуют десятки инструментов DevOps

Введение > CI > Среды > CD > CT > Метрики

1

Fm

Gh

Github

2

Fm

Aws

AmazonWeb Services

3

Os

Gt

Git

4

Pd

Dm

DBmaestro

11

Fm

Bb

Bitbucket

12

Os

Lb

Liquibase

19

Os

Gl

GitLab

20

En

Rg

Redgate

21

Os

Mv

Maven

22

Os

Gr

Gradle

23

Os

At

ANT

24

Os

Fn

FitNesse

25

Fr

Se

Selenium

26

Os

Ga

Gatling

27

Fr

Dh

Docker Hub

28

Os

Jn

Jenkins

29

Pd

Ba

Bamboo

30

Os

Tr

Travis CI

31

Pd

Gd

Deployment Manager

32

Os

Sf

SmartFrog

33

Os

Cn

Consul

34

Os

Bc

Bcfg2

35

Os

Mo

Mesos

36

En

Rs

Rackspace

37

Os

Sv

Subversion

38

En

Dt

Datical

39

Os

Gt

Grunt

40

Os

Gp

Gulp

41

Os

Br

Broccoli

42

Fr

Cu

Cucumber

43

Os

Cj

Cucumberjs

44

Fr

Qu

Qunit

45

Os

Npm

npm

46

Fm

Cs

Codeship

47

Pd

Vs

Visual Studio

48

Fm

Cr

CircleCI

49

Fr

Cp

Capistrano

50

Fr

Ju

JuJu

51

Os

Rd

Rundeck

52

Os

Cf

CFEngine

53

Fr

Ds

Swarm

54

Os

Op

OpenStack

55

Os

Hg

Mercurial

56

En

Dp

Delphix

57

Fr

Sb

sbt

58

Os

Mk

Make

59

Os

Ck

CMake

60

Fr

Ju

JUnit

61

Fr

Jm

JMeter

62

Fr

Tn

TestNG

63

Os

Ay

Artifactory

64

Fm

Tc

TeamCity

65

Fm

Sh

Shippable

66

Os

Cc

CruiseControl

67

En

Ry

RapidDeploy

68

Fm

Cy

CodeDeploy

69

En

Oc

Octopus Deploy

70

En

No

CA Nolio

71

Os

Kb

Kubernetes

72

Fm

Hr

Heroku

73

En

Cw

ISPW

74

En

Id

Idera

75

Os

Msb

MSBuild

76

Os

Rk

Rake

77

Fr

Pk

Packer

78

Os

Mc

Mocha

79

En

Xltv

XL TestView

80

Os

Jm

Jasmine

81

Os

Nx

Nexus

82

Os

Co

Continuum

83

Fm

Ca

Continua CI

84

Pd

So

Solano CI

85

En

Xld

XL Deploy

86

En

EB

ElectricBox

87

Fm

Dp

Deploybot

88

En

Ud

UrbanCode Deploy

89

Os

Nm

Nomad

90

En

Os

OpenShift

91

En

Xlr

XL Release

92

En

Ur

UrbanCode Release

93

En

Bm

BMC Release Process

94

En

Hp

HP Codar

95

En

Au

Automic

96

En

Pl

Plutora Release

97

En

Sr

Serena Release

98

Pd

Tfs

Team Foundation

99

Fm

Tr

Trello

100

Pd

Jr

Jira

101

Fm

Rf

HipChat

102

Fm

Sl

Slack

103

Fm

Fd

Flowdock

104

Pd

Pv

Pivotal Tracker

105

En

Sn

ServiceNow

106

Os

Ki

Kibana

107

Fm

Nr

New Relic

108

Os

Ni

Nagios

109

Os

Zb

Zabbix

110

En

Dd

Datadog

111

Os

El

Elasticsearch

112

Os

Ss

StackState

113

En

Sp

Splunk

114

Fm

Le

Logentries

115

Fm

Sl

Sumo Logic

116

Os

Ls

Logstash

117

Os

Gr

Graylog

118

Os

Sn

Snort

119

Os

Tr

Tripwire

120

En

Ff

Fortify

Os

Open Source

Fr

Free

Fm

Freemium

Pd

Paid

En

Enterprise

SCM

CI

Deployment

Cloud / IaaS / PaaS

BI / Monitoring

Database Mgmt

Repo Mgmt

Config / Provisioning

Release Mgmt

Logging

Build

Testing

Containerization

Collaboration

Security

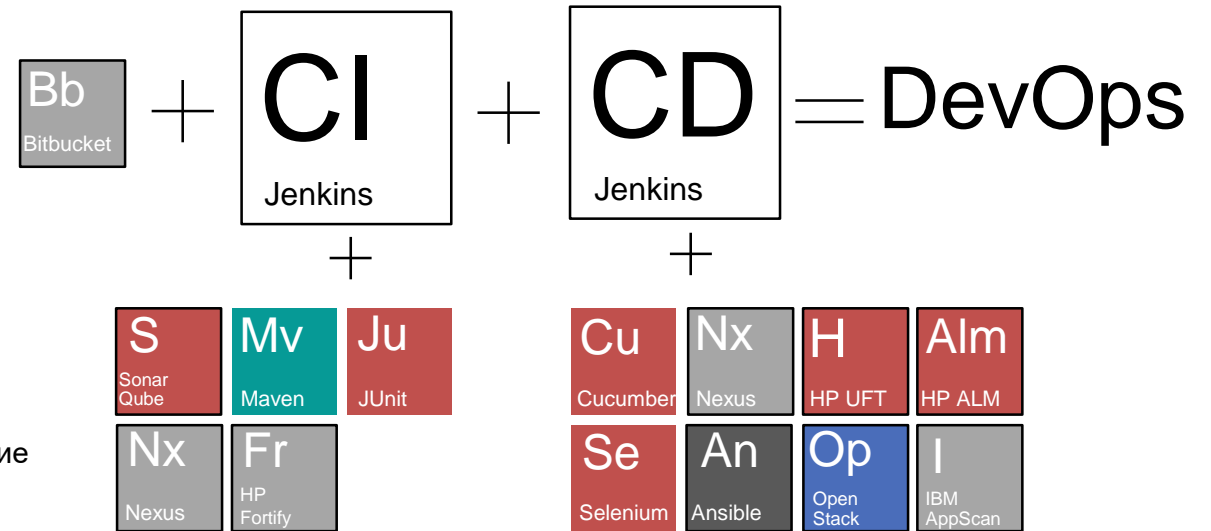
Источник: <https://xebialabs.com/periodic-table-of-devops-tools/>

Инструменты: утвержденные и выбираемые командами

Введение > CI > Среды > CD > CT > Метрики

Утвержденные инструменты:

- **Atlassian Bitbucket** – система версионного хранения кода
- **Maven** – инструмент авто-сборки
- **JUnit** – библиотека для модульного тестирования Java приложений
- **Jenkins** – инструмент управления процессом сборки, развертывания и настройки
- **Sonar, HP Fortify, IBM AppScan** – проверка кода на качество и уязвимости
- **HP ALM** – инструмент управления процессом тестирования
- **HP UFT, Selenium, Cucumber, HP PC** – автотесты, нагрузочное тестирование
- **Ansible** – управление конфигурациями и настройкой сред
- **Nexus** – управление хранением дистрибутивов
- **OpenStack** – виртуализация инфраструктуры, частное облако



□ - Целевой инструмент, утвержденный / рекомендованный Архсоветом

Инструменты, выбираемые командами:

Однако существуют инструменты, тесно связанные с определенной технологией. Такие инструменты не утверждаются на архитектурном совете, их выбор осуществляется рабочей группой:

- Командой разработки
- ЦЭ процессов производства (ЦЭ DevOps)
- Командой сервиса CI или CD
- Прикладными и системными администраторами

Почему нам важно иметь стандарты в выборе инструментов DevOps?

Введение > CI > Среды > CD > CT > Метрики

- Централизованное сопровождение инструментов позволяет командам сосредоточиться на разработке
- Предоставление централизованного сервиса 24x7
- Объективный выбор инструментов с учетом критериев стоимости (в т.ч. стоимость эксплуатации), эффективности и надежности и соответствие инструментов принципам прозрачности
- Наличие множества инструментов*, выполняющих одинаковую функцию, создает «бутылочное горлышко» на стороне Ops, что противоречит принципам DevOps.
- К централизованным инструментам прилагается централизованный мониторинг, который позволяет проследить весь путь релиза от разработки до ПРОМа.

*Проблема множества инструментов усугубляется тем, что в эксплуатации могут находиться несколько версий одного и того же инструмента

DevOps Toolchain

Непрерывная интеграция

1. Кодирование



Разработчик программирует код и пишет модульные тесты к этому коду

Инструменты: Bitbucket JUnit

2. Управление сборкой



После сохранения кода в **Bitbucket** запускается workflow процесса сборки в **Jenkins**. **Jenkins** запускает модульные тесты, статический анализ качества кода и ИБ. **Ansible** производит автоматизированное развёртывание дистрибутива на среду разработки.

Инструменты: Nexus Bitbucket

Инструменты сборки: Jenkins ANSIBLE FORTIFY sonarqube

Непрерывная поставка

3. Подготовка среды



После сборки дистрибутива запускается workflow процесса развёртывания в **Jenkins**, происходит (опционально) выделение тестовых сред, подготавливаются тестовые наборы данных, осуществляется автоматическое конфигурирование сред через **Ansible**.

Инструменты:



Nexus

4. Управление развёртыванием



Ansible развёртывает дистрибутив на тестовые среды, выполняются технические и функциональные тесты развёртывания

Инструменты:



Nexus

5. Управление тестированием



Jenkins вызывает **HP ALM** и запускается набор тестов, как автоматизированных (API, GUI, интеграционных, безопасности и т.д.), так и «ручных»

Инструменты:



ALM



Unified Functional Testing (UFT)

AppScan

Введение > CI > Среды > CD > CT > Метрики

Мониторинг и обратная связь



Механизм оповещений и мониторинга гарантирует быстрый цикл обратной связи. Доступен для всех этапов

Инструменты:



elastic



QlikView

Непрерывное развёртывание

6. ПСИ



В случае успешного прохождения испытаний принимается решение (человеком) о развёртывании дистрибутива на ПРОМ среду

7. Развёртывание на ПРОМ



Происходит автоматизированное развёртывание дистрибутива на ПРОМ среду. В случае ошибок – выполняется автоматический откат (Rollback)

Инструменты:



Nexus

Новая роль – DevOps инженер

Введение > CI > Среды > CD > СТ > Метрики



DevOps инженер – является экспертом по практикам DevOps, владеет инструментами CI и CD, выстраивает Pipeline сборки и развертывания.

- Настраивает процесс CI на централизованном сервисе, интегрирует его с GitFlow и кодом развертывания на средах разработки.
- Разрабатывает процесс поставки ПО на целевых инструментах CD для всех сред (разработки, тестирования и эксплуатации).
- Обеспечивает единообразие процессов поставки и выстраивания их в единый бесшовный процесс поставки.
- Управляет и обеспечивает процесс распространения конфигураций тестовых сред для сред разработки и тестирования (за границами обязанностей среды Банка).
- Оказывает сопровождение прикладным и системным администраторам сред Банка
- Владеет инструментами централизованного сервиса CI и CD, а также командными инструментами, в т.ч. по юнит-тестированию и сборки.
- Обладает знаниями технической и интеграционной архитектуры ОРД и механизмов переносов настроек между средами.
- Обладает базовыми навыками администрирования прикладного и системного ПО ОРД.

Работа DevOps Toolchain

Введение > CI > Среды > CD > CT > Метрики

