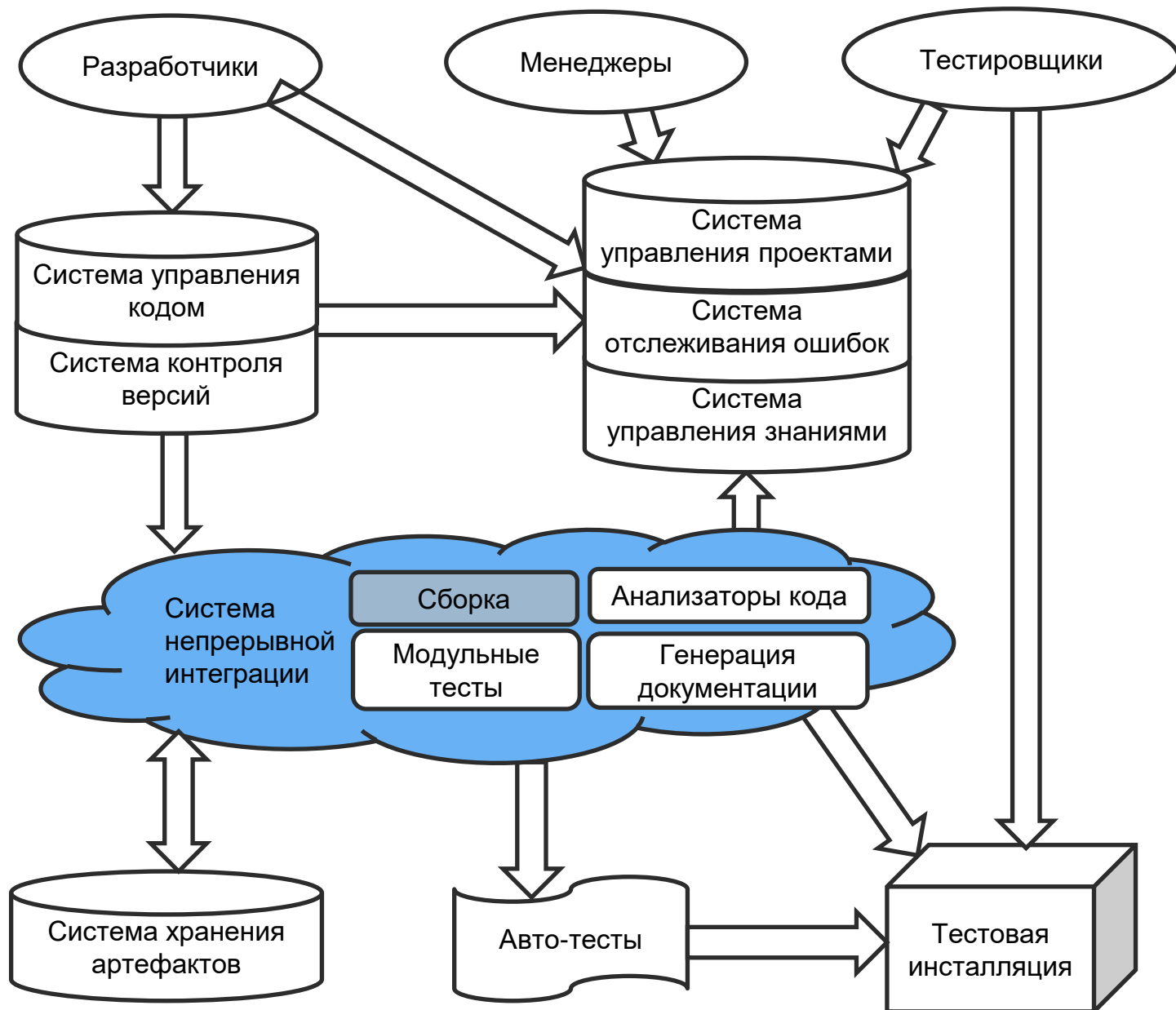


Управление производственным процессом разработки программного обеспечения

Системы сборки и Continuous Integration



План

- Системы сборки первого поколения (Make)
- 40 лет эволюции (Autotools)
- Сборка в мире Java (Ant)
- New age builders (Maven)
- Continuous integration
- CI в кросс-платформенных проектах

Сборка (building)

- Повторяемость. Проект должен собираться одним способом у всех участников.
- Простота. Новые участники должны иметь возможность собирать проект с минимумом дополнительных затрат.
- Управление зависимостями
- Запуск автоматических тестов

Артефакты

- Исполняемые файлы
- Документация
- jar/war в Java
- APK в Android
- IPA в iOS
- Инсталлятор в Windows
- deb/rpm пакет в Linux
- прошивки для устройств

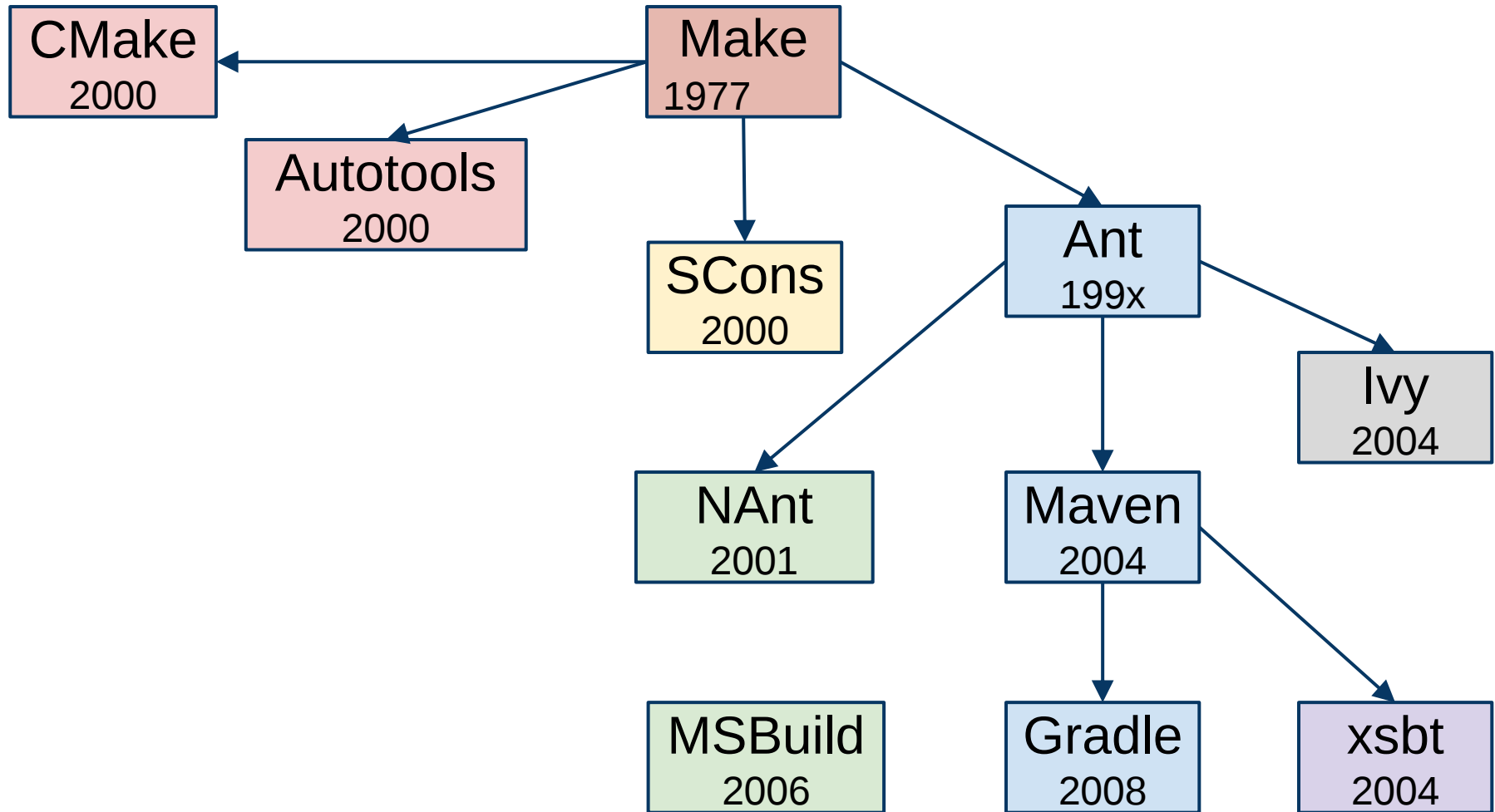
Артефакты в вебе

- Архив с файлами для развертывания системы
- Сама развернутая система

Задачи сборки

- Компиляция
- Генерация документации
- Сборка исполняемого файла
- Сборка инсталлятора
- Упаковка ресурсов (склейка картинок)
- Перпроцессинг (SASS, LESS, CoffeeScript)
- Разрешение зависимостей
- Прогон unit-тестов
- Публикация результатов

Основные системы сборки*



*) даты приблизительные

Make

```
CC      = gcc
CFLAGS  = -g
```

```
all: helloworld
```

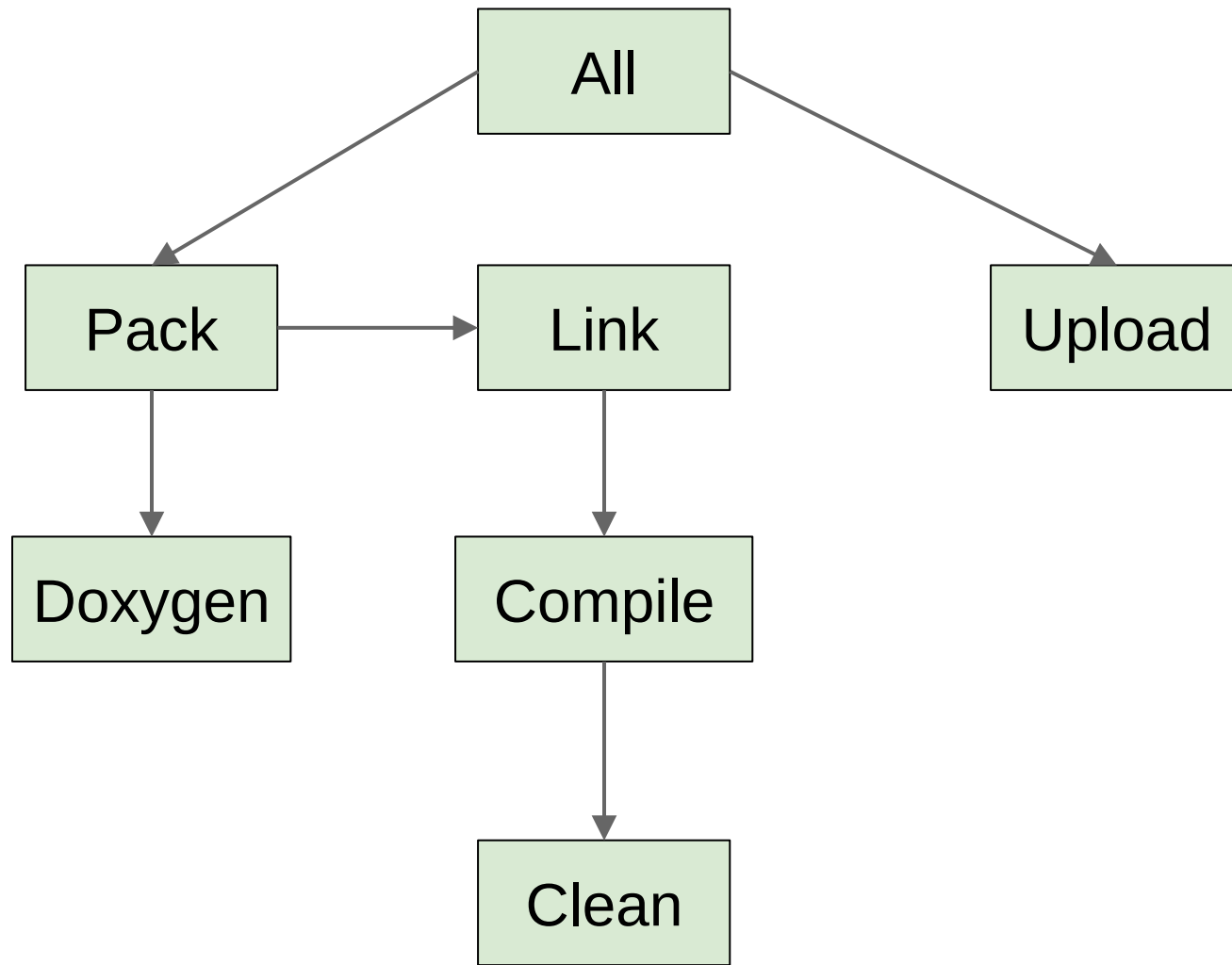
```
helloworld: helloworld.o
    $(CC) $(LDFLAGS) -o $@ $^
```

```
helloworld.o: helloworld.c
    $(CC) $(CFLAGS) -c -o $@ $<
```

```
clean: FRC
    rm -f helloworld helloworld.o
```

```
FRC:
```

Задачи и зависимости



Недостатки Make

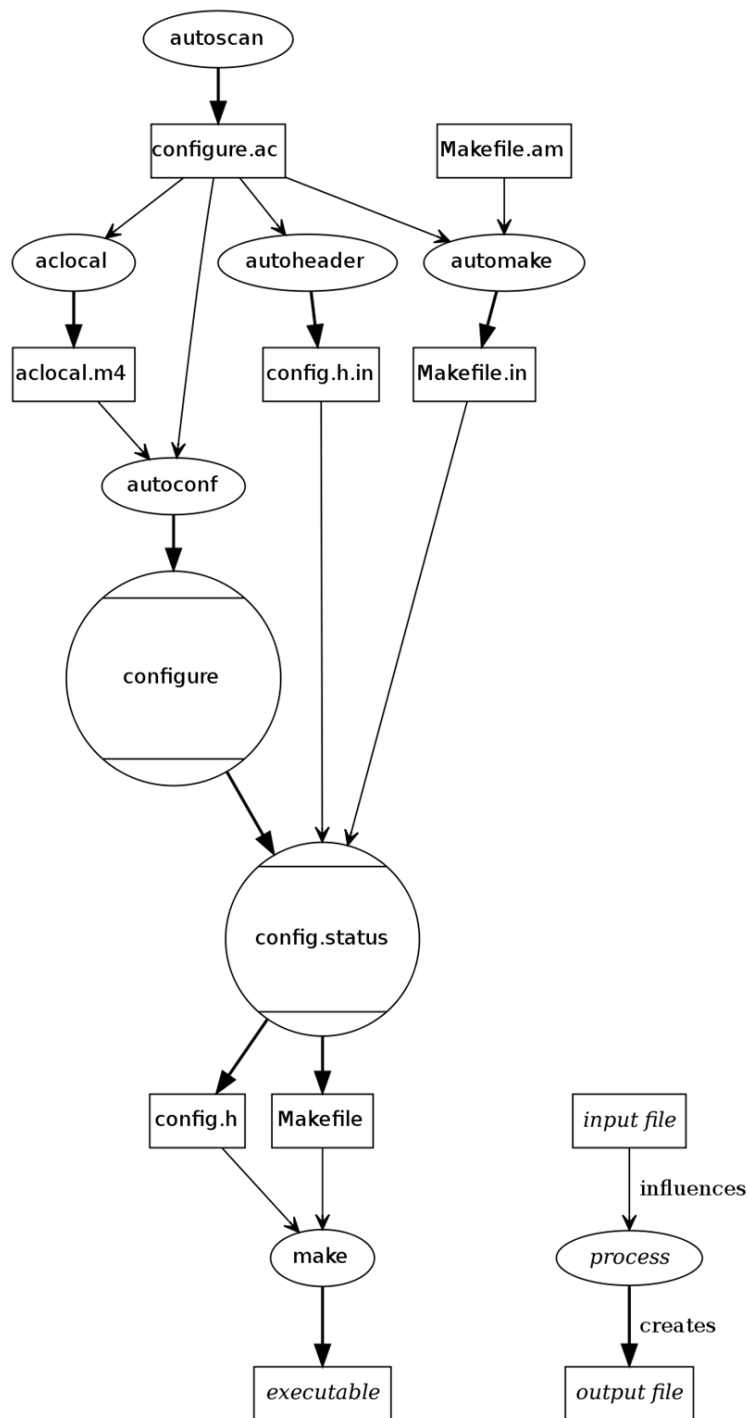
- Makefile получается очень большим
- Makefile сложно читать
- Необходимо часто изменять Makefile
- Нет управления зависимостями
- Нет инкрементальной компиляции

Autotools

- Автоматическая генерация Makefile
- Макросы m4
- Сборка проектов сводится к запуску:

```
./configure  
make  
make install
```

- Скрипт configure генерируется с помощью autoconf
- За долгие годы внутри autotools накопилась черная магия



Эпоха экспериментов

- SCons (Python)

```
env = Environment()  
env.Append(CPPFLAGS=['-Wall', '-g'])  
env.Program('hello',  
            ['hello.c', 'main.c'])
```

- CMake (препроцессор для Make)

```
cmake_minimum_required (VERSION 2.6)  
project (Tutorial)  
add_executable(hello.c)
```

Тем временем в другой галактике...

Apache Ant (Java)

- Тот же Make но в “современном” XML синтаксисе:

```
<target name="build" depends="clean">
    <mkdir dir="${build.classes}"/>

    <javac srcdir="${src.dir}"
          destdir="${build.classes}">
        <classpath refid="libs.dir"/>
    </javac>

    <jar jarfile="${build}/${name}.jar">
        <fileset dir="${build.classes}"/>
    </jar>
</target>
```

```
#ant build
```

Apache Ant (Java)

- Декларативный синтаксис
- XML
- Нет управления зависимостями
- Нет инкрементальной компиляции
- Достаточно гибкий для любых проектов
- Существует порт для .NET (NAnt)
- Не прижился за пределами Java экосистемы

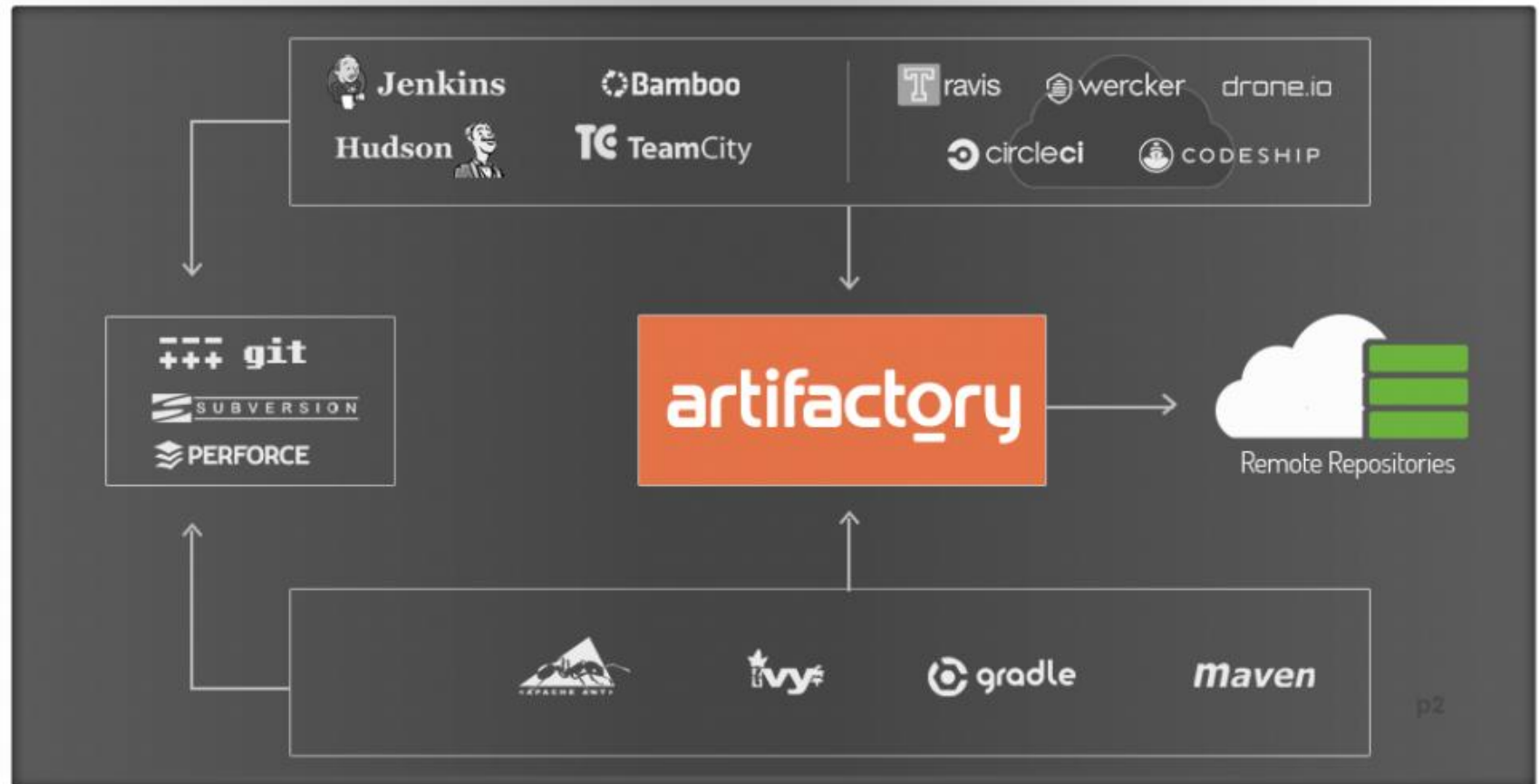
Управление зависимостями

Большинство проектов использует сторонние библиотеки которые нужно где-то хранить. Эти библиотеки могут зависеть от других библиотек и эти зависимости нужно как-то разрешать. В Java эта проблема носит название `classpath hell`.

Стандартные решения:

- Оставить эту проблему на откуп желающим собрать проект.
- Хранить зависимости в VCS.
- Использовать универсальное хранилище (FTP, Shared folder, HTTP)
- Использовать специализированное хранилище.

Artifactory



Apache Maven

- Convention over configuration
- Стандартная структура папок:

```
/src
  /main
    /java
      /ru/nsu/...
    /resources
    /config
    /webapp
  /test
    /java
    /resources
```

README

LICENSE

pom.xml

- Управление зависимостями

pom.xml

```
<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>ru.nsu.fit</groupId>
  <artifactId>hello-world</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.2</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

```
#mvn package
```

Управление зависимостями

В настройках maven указываются репозитории из которых разрешаются зависимости. При сборке maven пытается разрешить зависимости из кэша, локального репозитория и, наконец, из центрального репозитория. После сборки артефакт помещается в локальный репозиторий.

<http://search.maven.org>

Maven умеет выкачивать исходники и документацию зависимых библиотек и генерировать проектные файлы для IDE.

```
#mvn idea:idea
```

Недостатки Maven

- Очень плохо кастомизируется
- Все что лежит за пределами стандартного workflow настраивается довольно сложно
- Провоцирует неряшливое управление зависимостями

Gradle

- Используем Maven репозиторий
- Пишем сборку на Groovy
- Набор стандартных плагинов

Pros:

- Компактная запись по сравнению с XML
- Расширяем build script быстро и в любую сторону

Cons:

- Долгий старт
- Постоянно «стреляем себе в ногу»

Gradle

```
apply plugin: 'java'
apply plugin: 'maven'
apply plugin: 'eclipse'

version = System.getProperty('version', 'development')

ext {
    commit = System.getProperty('commit', 'unknown')

    jacksonVersion = '2.8.8'
    logbackVersion = '1.1.3'
}

sourceCompatibility = "1.8"
targetCompatibility = "1.8"

dependencies {
    compile "org.slf4j:slf4j-api:1.7.12",
            "org.apache.commons:commons-lang3:3.4",
            "commons-collections:commons-collections:3.2.1",
            "com.google.guava:guava:18.0",
            "ch.qos.logback:logback-classic:$logbackVersion",
            "ch.qos.logback:logback-core:$logbackVersion",
            "com.fasterxml.jackson.core:jackson-databind:$jacksonVersion",
            "com.fasterxml.jackson.core:jackson-annotations:$jacksonVersion"

    testCompile "junit:junit:4.12",
            "org.powermock:powermock-module-junit4:1.6.3",
            "org.powermock:powermock-api-mockito:1.6.3",
            "org.mockito:mockito-all:1.10.19"
}

repositories {
    mavenCentral()

    maven {
        name "Internal repository"
        url LOCAL_REPO
        credentials {
            username LOCAL_USER
            password LOCAL_PASSWORD
        }
    }
}
```

Gradle

```
task java2wsdl(dependsOn: project(":server").compileJava) {
    doLast {

        mkdir("$rootDir/server/flash/gen")

        println 'Generating Classes for use with WSDL'

        ant.xslt(
            classpath:
configurations.axisGenAntTask.plus(project(':server').configurations.compile).asPath,
            style: "$rootDir/server/xslt/extractMethods.xslt",
            in: "$rootDir/server/server-config.wsdd",
            out: "methods.tmp"
        )

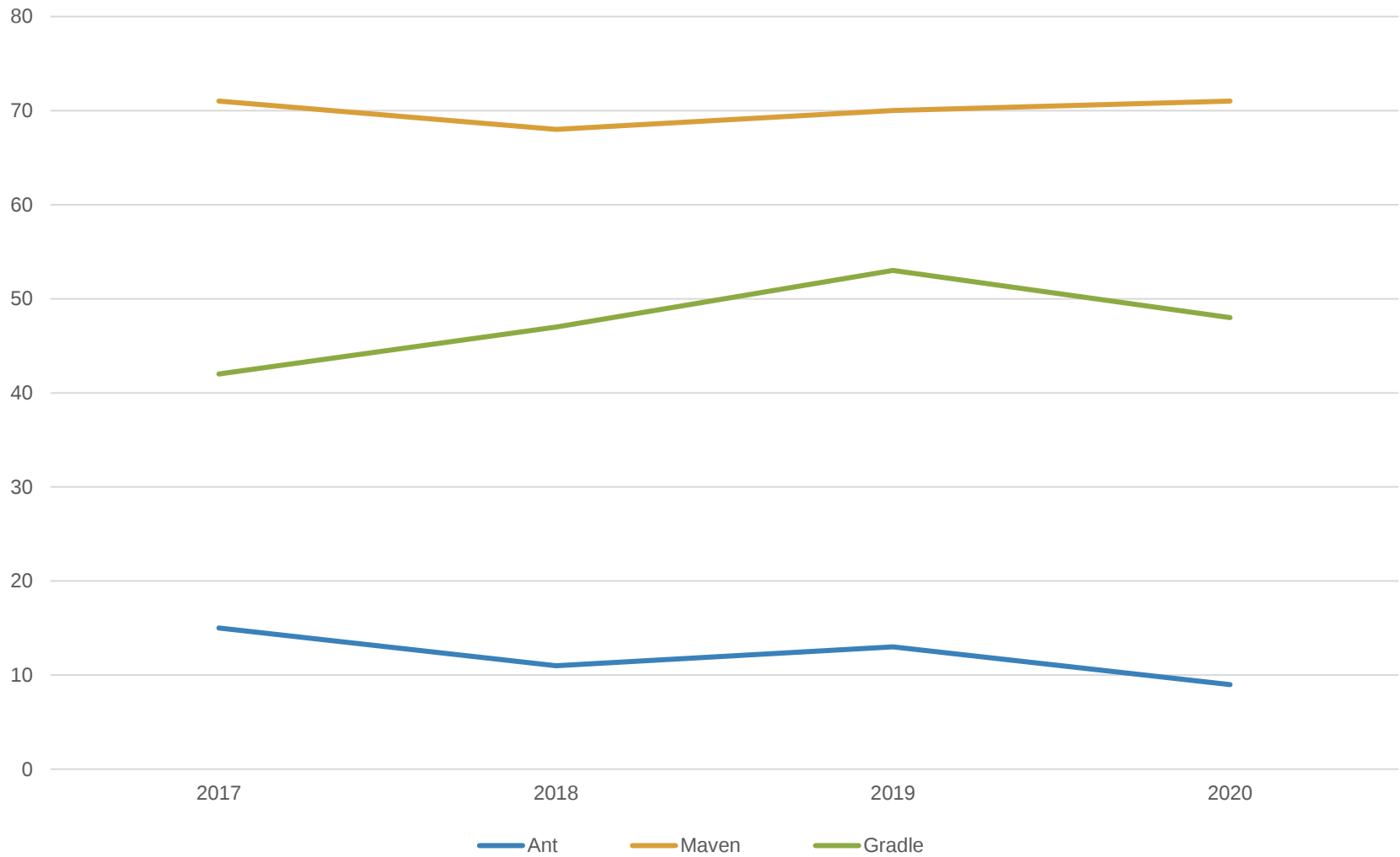
        ext.methodList = new Properties()
        methodList.load(file("$rootDir/methods.tmp").newDataInputStream())
        delete file("$rootDir/methods.tmp")

        ...
    }
}
```

Gradle

```
task dist(type: Tar, dependsOn: [jar, apiDocs, createData, copyScripts]) {  
    compression = Compression.GZIP  
    archiveName = "server-${version}.tar.gz"  
    from("$buildDir/output") {  
        include("server-$version/")  
    }  
}  
  
task copyFix(type: Copy) {  
    from configurations.tomcatFix  
    into "$buildDir/output/fixes_  
    rename("tomcat-fix-${tomcatFixVersion}.jar", "tomcat-fix.jar")  
}
```

Usage in % of interviewed developers



Источник: <https://www.jetbrains.com/lp/devecosystem-2020/>

Другие языки

- Python: SetupTools
- Ruby: Rake
- .NET: MSBuild
- Groovy: Gradle, Grape
- Scala: SBT

Continuous Integration

Сборочный сервер

- Унификация среды
- Выделенный сервер с установленным набором необходимых библиотек, на котором выполняется сборка.
- Сборка по необходимости
- Сборка по расписанию (ночные сборки)
- Сборка по изменению исходного кода (CI)

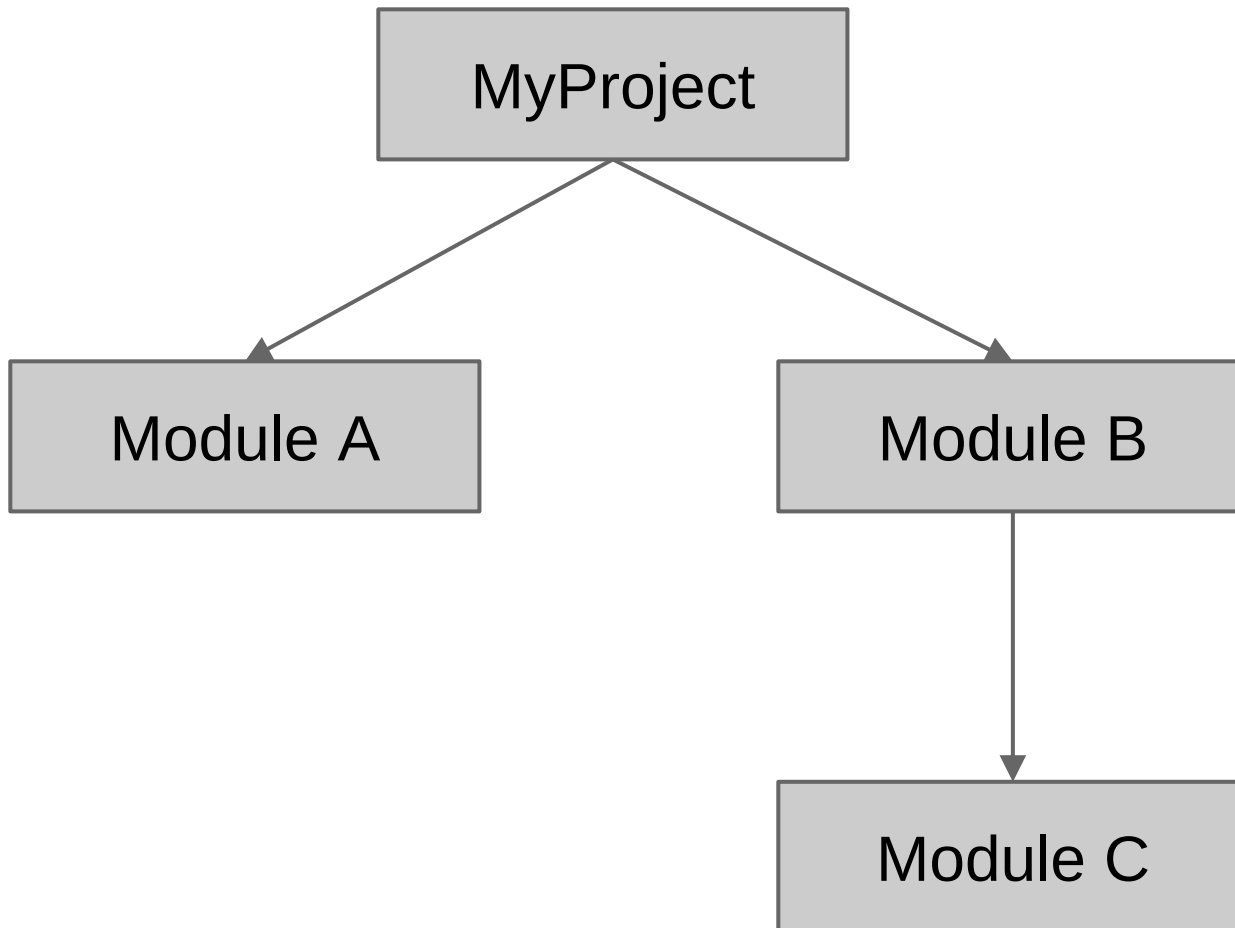
Преимущества CI

- Эталонная среда сборки
- Раннее обнаружение проблем интеграции
- Постоянное наличие текущей версии продукта
- Дисциплинирует разработчиков
- Позволяет отслеживать метрики в реальном времени
- Немедленный прогон тестов после каждого изменения
- Разгружает машину разработчика от прогона тестов и сборки

Задачи CI сервера

- Генерация артефактов
- Прогон тестов
- Нотификация разработчиков о неудаче
- Сбор метрик (тестовое покрытие)
- Статический анализ кода
- Сборка проекта из зависимых модулей
- Кросс-платформенная сборка

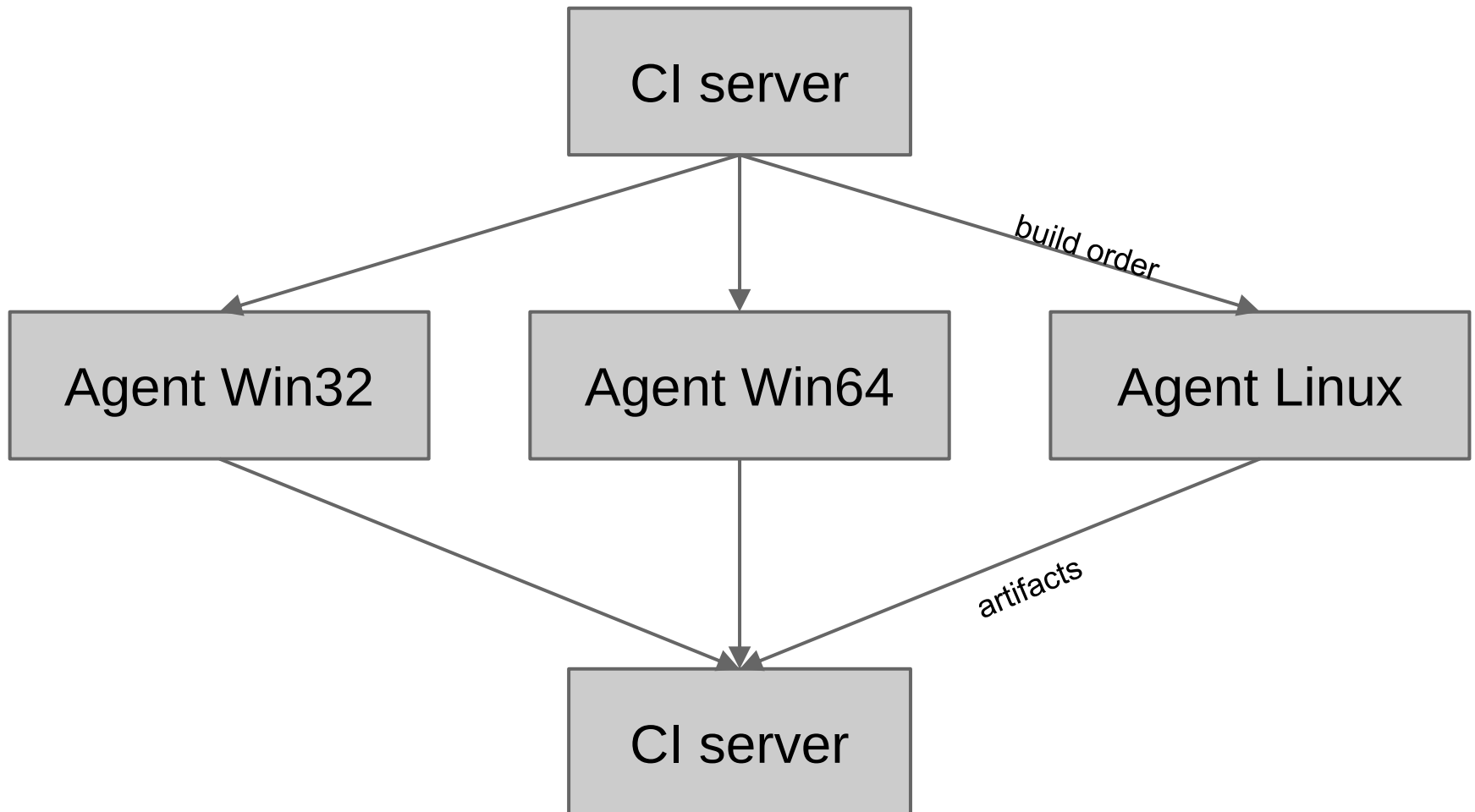
Зависимые модули



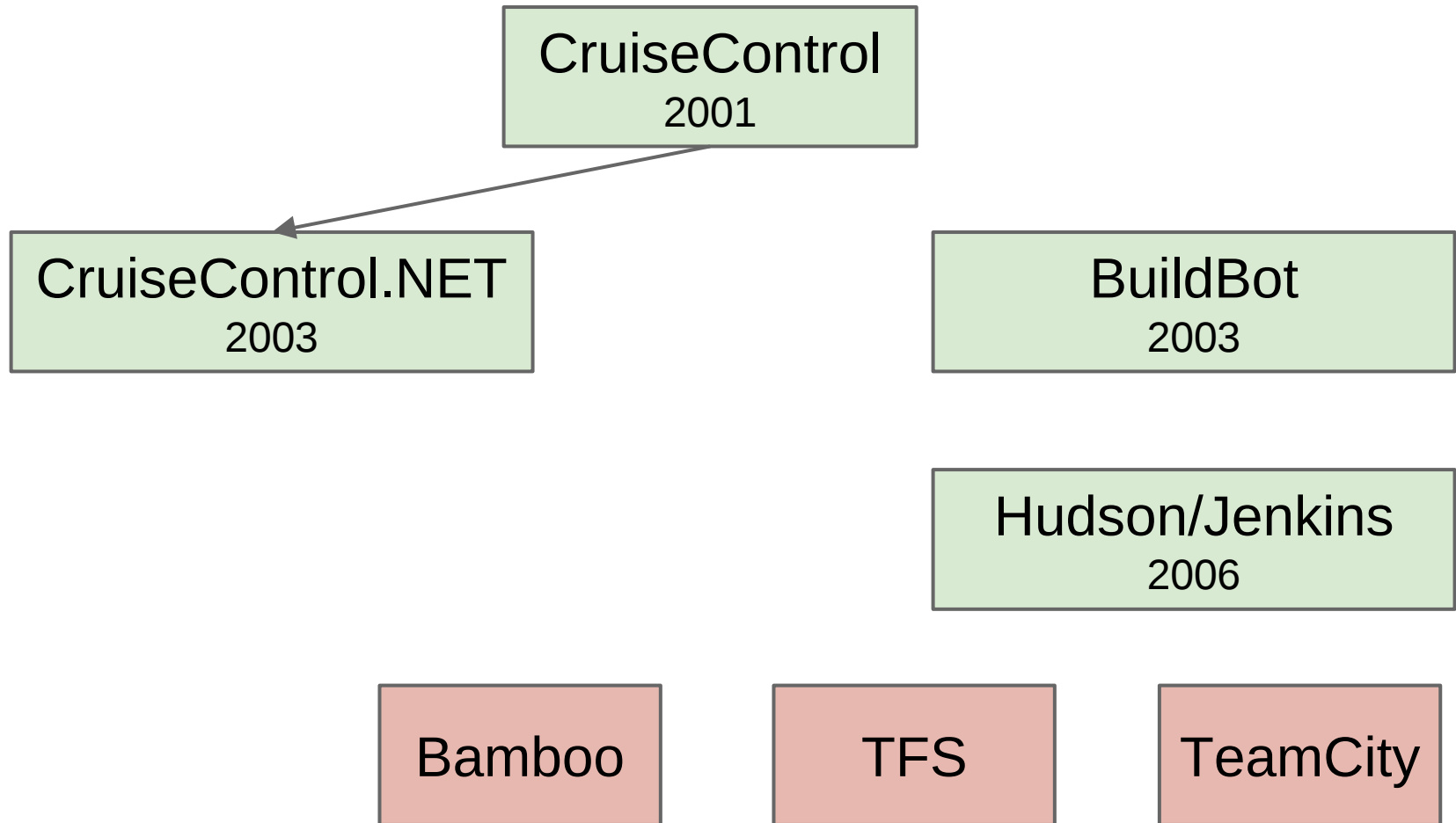
Кросс-платформенная сборка

- Кросс-компиляция
- Сборка в чужеродной среде

Агенты

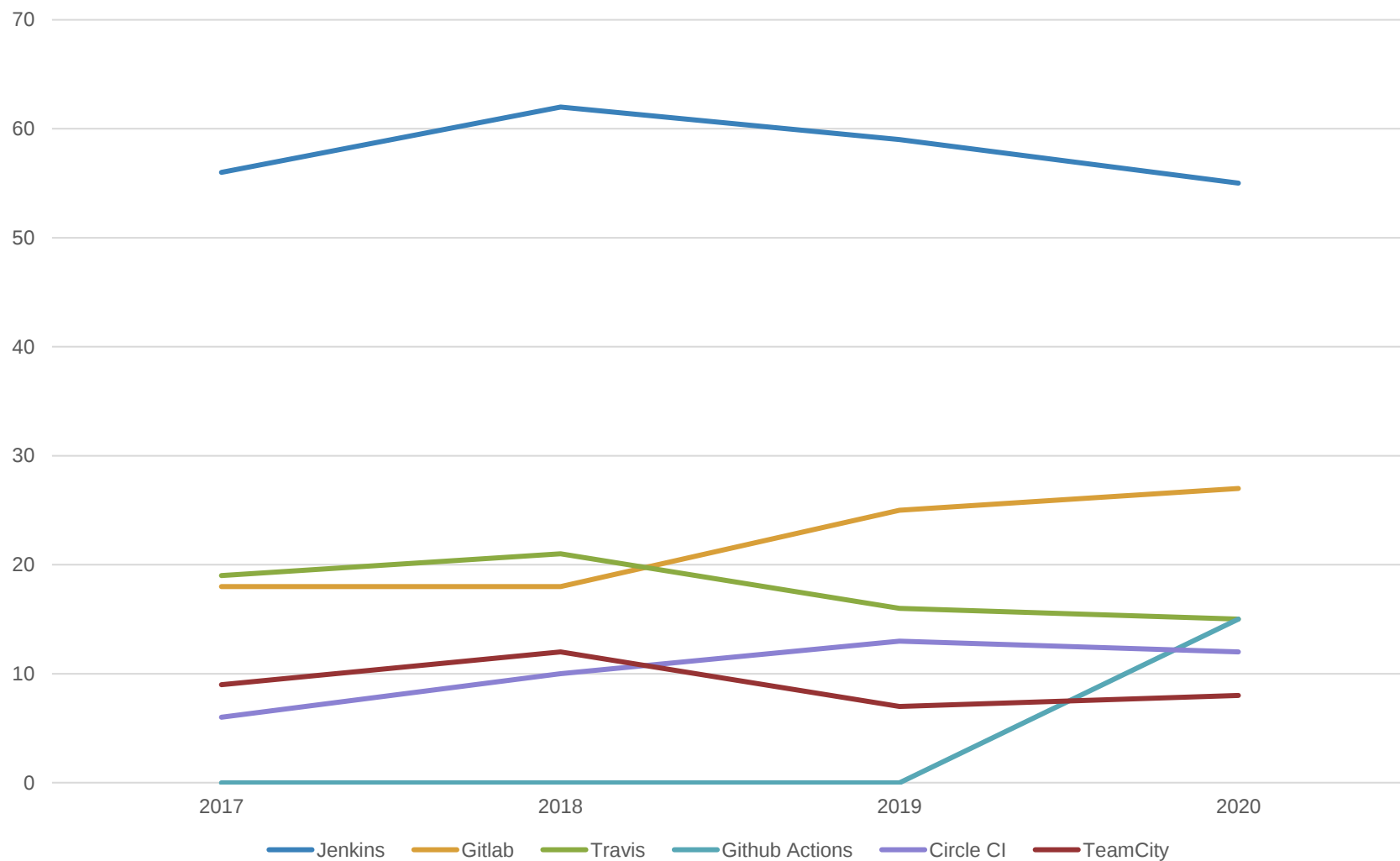


Основные системы CI*



**) даты приблизительные*

Usage in % of interviewed developers



Источник: <https://www.jetbrains.com/lp/devecosystem-2020/>

GitHub: интеграция с CI

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Sessions

Blocked users

Repositories

Organizations

Saved replies

Applications

Developer settings

Applications

Installed GitHub AppsAuthorized GitHub AppsAuthorized OAuth Apps

You have granted **1 application** access to your account. Sort ▾ Revoke all



Azure Pipelines

Last used within the last week · Owned by [AzurePipelines](#)

Revoke

ⓘ Read more about connecting with third-party applications at [GitHub Help](#).

GitHub: интеграция с CI

danoykin / test Private

Watch 0Star 0Fork 0

<> Code

Issues 2

Pull requests 1

Projects 1

Insights

Settings

#6 Add signature after the 'Hello World' message #7

Edit

Open

danoykin wants to merge 10 commits into master from issue/6-add-signature

Conversation 1

Commits 10

Checks 1

Files changed 6

+38 -38

6142565 — #6 Publish tests

1 successful check

Azure Pipelines

Succeeded — 3 minutes ago

danoykin.test (1)

danoykin.test (1)

Success

ran 3 minutes ago in less than a minute

6142565 by @danoykin

issue/6-add-signature

Build #20190125.11 succeeded

0 errors / 0 warnings

DETAILS

Tests

- 0 failed (0.00 %)
- 1 passed (100.00 %)
- 0 other (0.00 %)
- 1 total

GitHub: интеграция с CI

Open

#6 Add signature after the 'Hello World' message #7

Opened by danoykin

🔗	#6 Fix compilation issues	✗ 1703678
🔗	#6 Fix compilation issues	✗ ccc94ed
🔗	#6 Remove lombok	✗ 24f61a5
🔗	#6 Remove lombok	✓ 79c2f84
🔗	#6 Process tests	✓ a07749d
🔗	#6 Publish tests	✓ 6142565

Add more commits by pushing to the `issue/6-add-signature` branch on `danoykin/test`.



All checks have passed

1 successful check

[Show all checks](#)



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or [view command line instructions](#).



Write

Preview

AA B i “ < > 🔗 ☰ ☷ ☹ @ 📎 ↶

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

💡 Styling with Markdown is supported

Close pull request

Comment

💡 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.

GitHub: интеграция с CI

danoykin / test Private

Watch 0Star 0Fork 0

<> Code

Issues 2

Pull requests 0

Projects 1

Insights

Settings

Overview

Yours

Active

Stale

All branches

Search branches...

Default branch

master Updated 5 minutes ago by danoykin ✓ Default Change default branch

Your branches

issue/6-add-signature Updated 13 minutes ago by danoykin ✓ 1 | 0 #7 Merged 🗑

issue/1 Updated 4 hours ago by danoykin ✗ 14 | 0 #5 Merged 🗑

Active branches

issue/6-add-signature Updated 13 minutes ago by danoykin ✓ 1 | 0 #7 Merged 🗑

issue/1 Updated 4 hours ago by danoykin ✗ 14 | 0 #5 Merged 🗑

Bitbucket: Continuous integration

</> openrtb

<> Source

Commits

Branches

Pull requests

Pipelines

Deployments

Downloads

Boards

Settings

soft / rtb-libraries / openrtb

Pipelines

All branches

OTHER BRANCHES

master MAIN BRANCH

deploy-test

test/deploy

deploy

What's new

Schedules

Caches

Usage

?

Pipeline	Status	Started	Duration
#57 refactoring Alexey Emelyanov 892feea master	✓ Successful	a month ago	35 sec
#56 PR-555 Project migration to Java10 Alexey Emelyanov 841f7f3 master	! Failed	4 months ago	16 sec
#55 PR-555 Project migration to Java10 Alexey Emelyanov edb1a37 master	! Failed	4 months ago	17 sec
#54 make all classes final Alexey Emelyanov c1d6aff master	✓ Successful	4 months ago	26 sec
#53 make all classes final Alexey Emelyanov c1d6aff master	! Failed	4 months ago	22 sec
#52 refactoring Alexey Emelyanov 2c97504 master	✓ Successful	6 months ago	25 sec
#51 Merged in deploy-test (pull request #1) Deploy test Alexander Kascheev 9e2d81b master	✓ Successful	6 months ago	37 sec
#50 finished Alexander Kascheev 0a76f3c deploy-test	✓ Successful	6 months ago	18 sec
#49 test Alexander Kascheev 3a0fc57 deploy-test	✓ Successful	6 months ago	19 sec
#48 test Alexander Kascheev 56a141d test/deploy	✓ Successful	6 months ago	17 sec

Prev 1 2 3 4 5 6 Next

Continuous Integration system configuration

The screenshot shows the Azure DevOps interface for configuring a pipeline. The left sidebar contains navigation links: Overview, Pipelines (selected), Builds, Releases, Library, Task groups, and Deployment groups. The top navigation bar shows the user 'aden0795' and the path 'aden / Pipelines'. The main area is titled 'danoykin.test (1)' and has tabs for YAML, Variables, Triggers, and History. The 'YAML' tab is active, showing a 'Get sources' task with inputs 'danoykin/test' and 'master'. The right sidebar contains configuration fields: 'Name' (danoykin.test (1)), 'Agent pool' (Hosted Ubuntu 1604), and 'YAML file path' (azure-pipelines.yml). Each field has an information icon and a refresh button.

Azure DevOps

aden0795 / aden / Pipelines

Search

aden

Overview

Pipelines

Builds

Releases

Library

Task groups

Deployment groups

danoykin.test (1)

YAML Variables Triggers History

Save & queue Discard Summary Queue

Pipeline

Build pipeline

Get sources

danoykin/test master

Name *

danoykin.test (1)

Agent pool ⓘ | Pool information | Manage

Hosted Ubuntu 1604

YAML file path * ⓘ

azure-pipelines.yml

Project settings

Continuous Integration system: run tests

Azure DevOps

aden0795 / aden / Pipelines / Builds / danoykin.test (1) / #20190125.11

Search

aden

Overview

Pipelines

Builds

Releases

Library

Task groups

Deployment groups

Project settings

#20190125.11: #6 Add signature after the 'Hello World' message

Release All logs

Validation of 7 triggered today at 5:54 for GitHub targeting danoykin/test master

Logs Summary Tests

Summary ^

1 Run(s) Completed (1 Passed, 0 Failed)

1
Total tests
+1

1 Passed

0 Failed

0 Others

100%
Pass percentage
↑ 100%

10ms
Run duration ⓘ
↑ +10ms

Test run Column Options

Filter by test name

Test file Owner Outcome: **Aborted (+1)**

Hooray! There are no test failures.

Jenkins Dashboard

 **Jenkins**

поиск

 3  2  aden  выход

Dashboard

 Создать Item

 Пользователи

 История сборок

 Связи проектов

 Проверить хэш файла

 Настроить Jenkins

 My Views

 Lockable Resources

 New View

добавить описание

prod-uswildВсе+

S	W	Name ↓	Последний успех	Последняя неудача	Последняя продолжительность	
		hb-config-server	1 месяц 2 дня - log	Н/Д	32 секунд	
		hb-endpoint	1 месяц 2 дня - log	Н/Д	8.3 секунд	
		hb-ui	3 дня 18 часов - log	Н/Д	5.1 секунд	
		hb-ui-admin-front	2 месяца 6 дней - log	Н/Д	6.3 секунд	
		hb-ui-front	11 дней - log	Н/Д	4.4 секунды	
		hb-ui-lib-sample	6 месяцев 1 день - log	Н/Д	0.83 секунд	
		k8s	Н/Д	Н/Д	Н/Д	
		kino-test	Н/Д	Н/Д	Н/Д	
		playables	Н/Д	Н/Д	Н/Д	
		prod_deploy_config_server	1 год 7 месяцев - #5	1 год 11 месяцев - #2	55 секунд	
		prod_deploy_config_server-us	1 месяц 26 дней - #8	2 месяца 25 дней - #1	1 минута 19 секунд	
		prod_deploy_endpoint	2 месяца 28 дней - #35	7 месяцев 25 дней - #29	3 минуты 6 секунд	
		prod_deploy	1 месяц 2 дня -	2 месяца 25 дней -	1 минута 39 секунд	

Очередь сборки

Очередь сборки пуста

Состояние сборщиков

1 В ожидании

2 В ожидании

3 В ожидании

4 В ожидании

5 В ожидании

Building [wild-backend](#) » [Scan Multibranch Pipeline](#)

Jenkins Builds History

 **Jenkins**

ПОИСК

3

2

aden

ВЫХОД

Dashboard > hb-ui > master >

 Up

 Status

 Changes

 Собрать сейчас

 Смотреть конфигурацию

 Full Stage View

 Pipeline Syntax

 История сборок

Тренд ^

find

#107

31.01.2020 8:19

#106

29.01.2020 11:43

#105

20.05.2019 8:51

#104

20.05.2019 8:31

#103

20.05.2019 8:26

#102

17.05.2019 8:16

#101

14.05.2019 8:13

#100

14.05.2019 8:10

#99

14.05.2019 7:45

#98

08.05.2019 7:00

#97

08.05.2019 6:52

#96

08.05.2019 7:00

Pipeline master

Полное название проекта: hb-ui/master

 Recent Changes

Stage View

Average stage times:
(Average full run time: ~1min 52s)

	Declarative: Checkout SCM	Configure	Build	Publish
#107	3s	581ms	55s	28s
Jan 31 14:19 4 commits	1s	591ms	1min 26s	1min 34s
#106	21s	674ms	5s failed	48ms failed
Jan 29 17:43 9 commits	1s	596ms	1min 41s	45s
#105	470ms	545ms	49s	3s failed
May 20 13:51 1 commit	879ms	546ms	2s failed	24ms failed
#104	1s	549ms	1min 50s	
May 20 13:31 1 commit				
#103				
May 20 13:26 1 commit				
#102				
May 17 13:16 3 commits				

Jenkins Advanced Build

 **Jenkins**

3

ПОИСК

aden-asteriosoft | Выход

Jenkins > _prod_us > deploy-bidders-selected >

 Up

 Статус

 Изменения

 Сборочная директория

 Собрать с параметрами

 Удалить Проект

 Настройки

 Move

 Rename

 **История сборок** [Тренд](#)

find x

#151819.03.2021 13:55

#151718.03.2021 10:19

#151617.03.2021 8:55

#149502.03.2021 18:14

 [RSS для всех](#)  [RSS для неудачных](#)

Проект deploy-bidders-selected

Для этой сборки необходимы следующие параметры:

doForAllInstances

If false: will execute only for selected hosts with Serial behavior
If true: ignore selected hosts and apply for all hosts with Serial behavior

bidders

☐ (172.31.18.1 - 3.22.209.54): release-1.245.1
☐ (172.31.28.113 - 13.58.111.30): release-1.245.1
☐ (172.31.27.59 - 18.191.51.96): release-1.245.1
☐ (172.31.27.81 - 3.135.222.168): release-1.245.1
☐ (172.31.29.24 - 3.17.188.183): release-1.245.1
☐ (172.31.27.27 - 18.220.248.35): release-1.245.1
☐ (172.31.27.28 - 3.17.58.199): release-1.245.1
☐ (172.31.26.86 - 18.191.250.56): release-1.245.1
☐ (172.31.30.7 - 3.138.199.31): release-1.245.1
☐ (172.31.26.164 - 3.135.234.27): release-1.245.1
☐ (172.31.19.115 - 3.142.151.144): release-1.245.1

Filter

ansibleSerial

Example: 1,2,3,5,25%
Run group tasks by instance's count:
- first on only one
- second - on two instances
- third - on three instances
- on five instances
- on each 25% instances

Собрать

Jenkins File

```
1  #!groovy  Anoykin, 15.12.2017 18:42 • PR-90 Prod cluster deployment
2
3  pipeline {
4      options {
5          disableConcurrentBuilds()
6          lock('bidder')
7      }
8      environment {
9          BUILD_VERSION = sh(
10             script: 'if [[ $BRANCH_NAME =~ "release-" ]]; then echo "${BRANCH_NAME}.${BUILD_ID}"; else echo "${BRANCH_NAME}.${BUILD_ID}-SNAPSHOT"; fi',
11             returnStdout: true
12         ).trim()
13     }
14     agent any
15     stages {
16         stage('Configure') {
17             steps {
18                 sh 'cp ./src/main/resources/application.yml.default ./src/main/resources/application.yml'
19                 sh 'cp ./src/main/resources/bootstrap.yml.qa ./src/main/resources/bootstrap.yml'
20             }
21         }
22         stage('Build') {
23             steps {
24                 script {
25                     sh "echo ${BRANCH_NAME} ${GIT_BRANCH} ${GIT_COMMIT} ${BUILD_VERSION}"
26                     sh "mvn clean package -Dbuild.version=${BUILD_VERSION}"
27                 }
28             }
29         }
30         stage('Publish') {
31             steps {
32                 script {
33                     if (env.BRANCH_NAME == 'master' || env.BRANCH_NAME.startsWith('release-') || env.BRANCH_NAME.startsWith('rc-')) {
34                         sh "echo ${BRANCH_NAME} ${GIT_BRANCH} ${GIT_COMMIT} ${BUILD_VERSION}"
35                         sh "mvn deploy -Dbuild.version=${BUILD_VERSION}"
36                     }
37                 }
38             }
39         }
40     }
41     post {
42         always {
43             junit 'target/surefire-reports/*.xml'
44         }
45     }
46 }
```