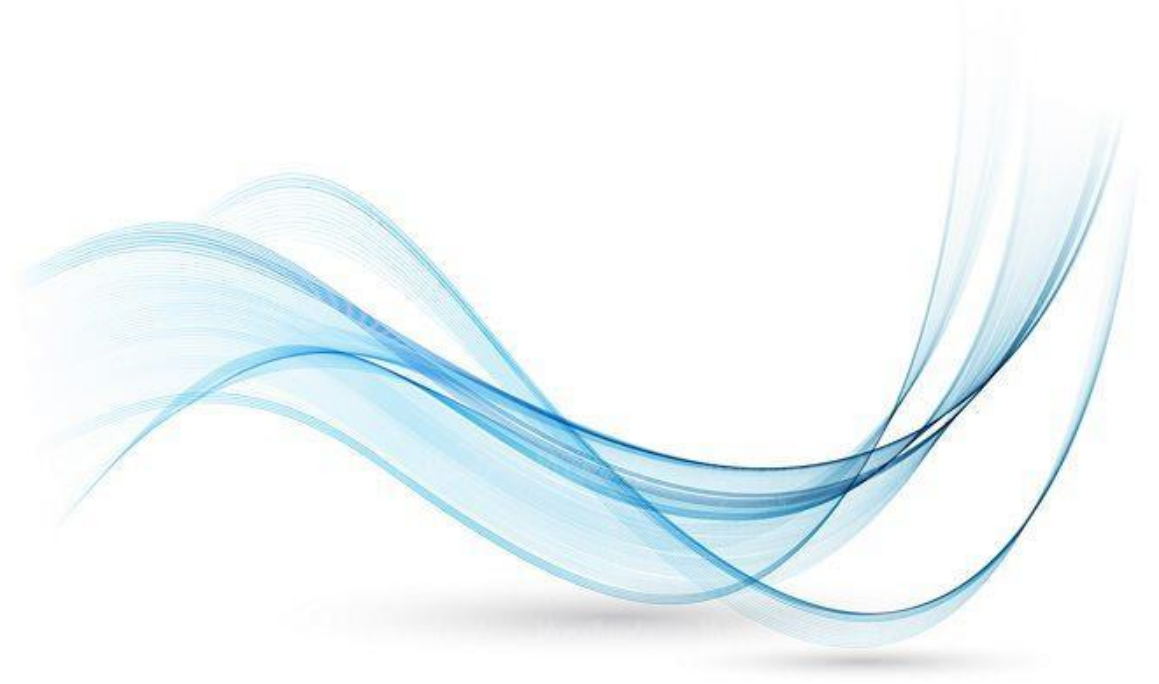


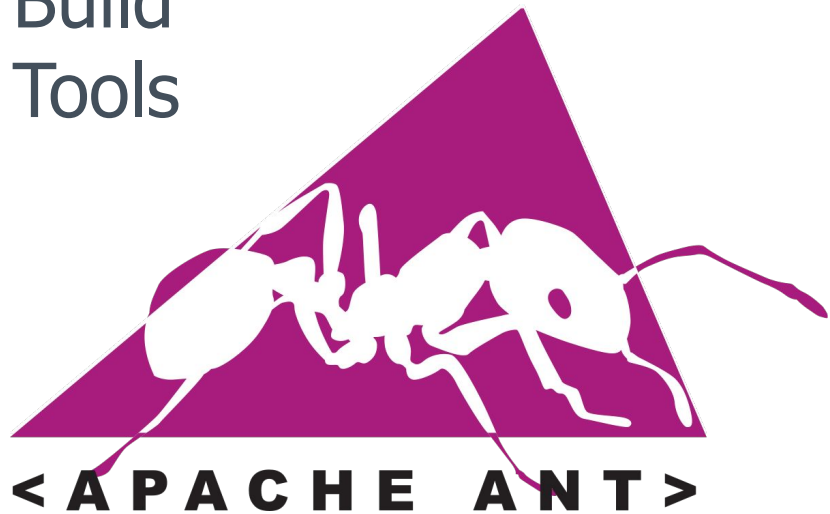
# Apache Maven

---

Evgeniy Shvetsov  
2021

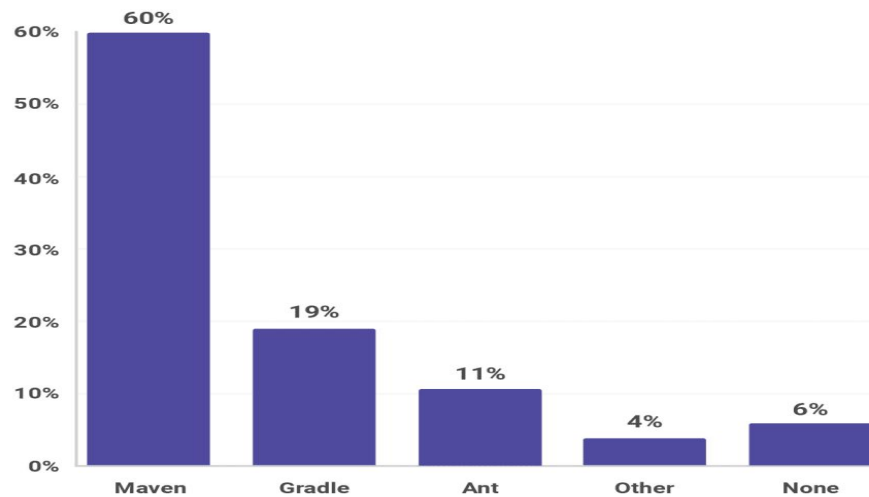


# Build Tools



Apache

**Maven™**



2018 - Which build tool do you use for your main project?

# Build Tools – Apache Maven

## Project Management Tool

**Первый выпуск:**

30 марта 2002 года

**Аппаратная платформа:**

JVM

**Управление проектом:**

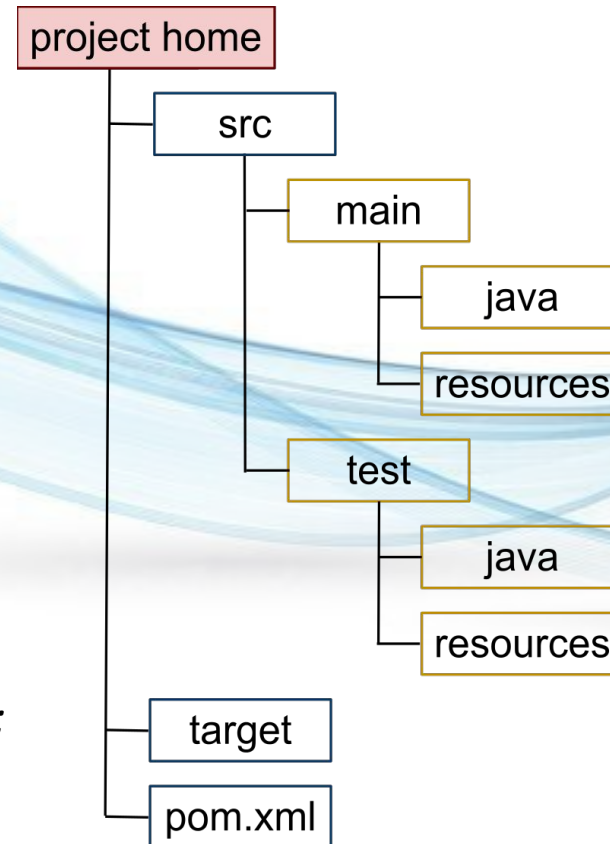
POM (Project Object Model) -  
XML-описание (pom.xml)

**Стиль:**

декларативный

**Основные концепции:**

- ✓ Соглашение по конфигурации;
- ✓ Архетипы - шаблоны;
- ✓ Жизненный цикл – деление на фазы;
- ✓ Архитектура – использование плагинов;
- ✓ Модули;
- ✓ Зависимости;
- ✓ Плагины;
- ✓ Профили.



**Достоинства:**

- ☐ Повторяемость сборки;
- ☐ Стандартизация сборки;
- ☐ Декларативность сборки;
- ☐ Автоматизация процесса сборки;
- ☐ Модульность;
- ☐ Управление зависимостями;
- ☐ Расширяемость плагинами.

**Недостатки:**

- ☐ Некоторая неэффективность для малых проектов.

# Maven



## GroupId

Это ID группы проекта. Зачастую, это уникальная организация или проект.



## ArtifactId

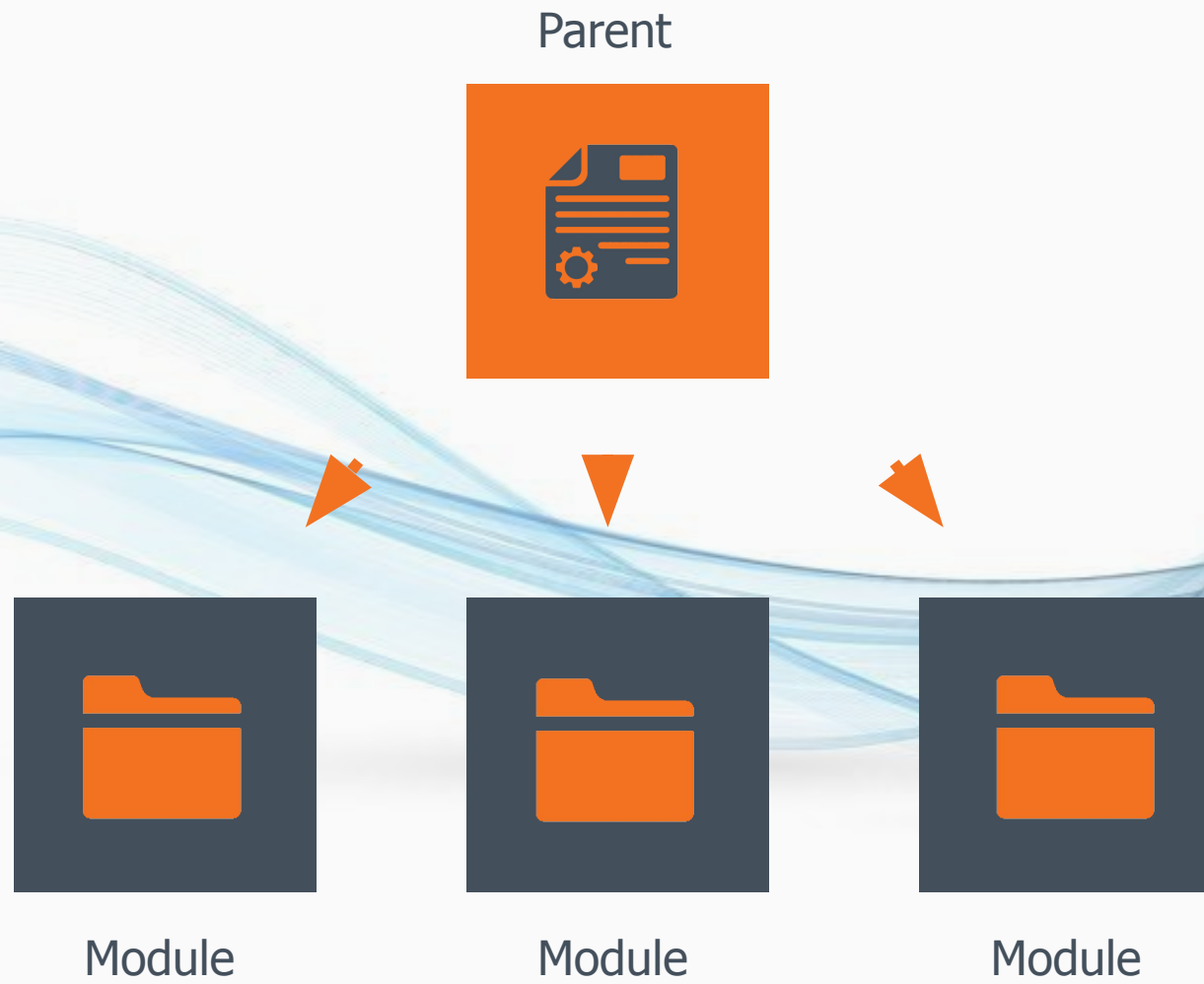
Это идентификатор самого проекта. Чаще всего – его имя



## Version

Версия проекта

# Maven



# Maven

## Основные фазы сборки проекта

### 1.compile

Компилирование проекта

### 2.test

Тестирование с помощью JUnit тестов

### 3.package

Создание .jar файла или war, ear в зависимости от типа проекта

### 4.integration-test

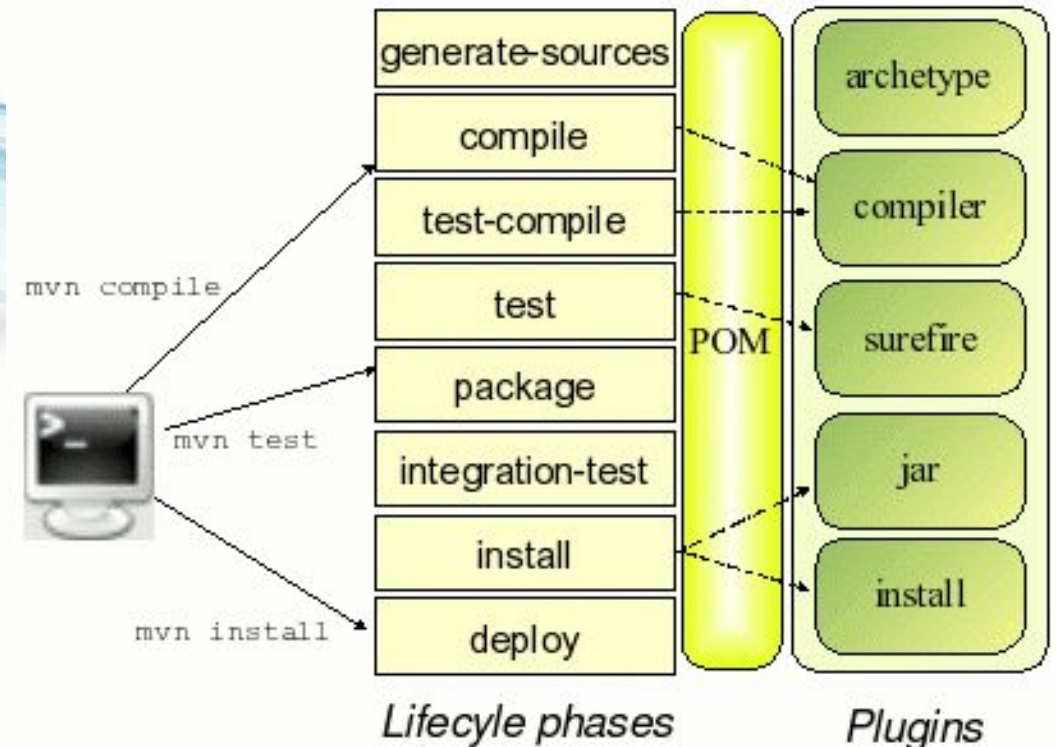
Запуск интеграционных тестов

### 5.install

Копирование .jar (war , ear) в локальный репозиторий

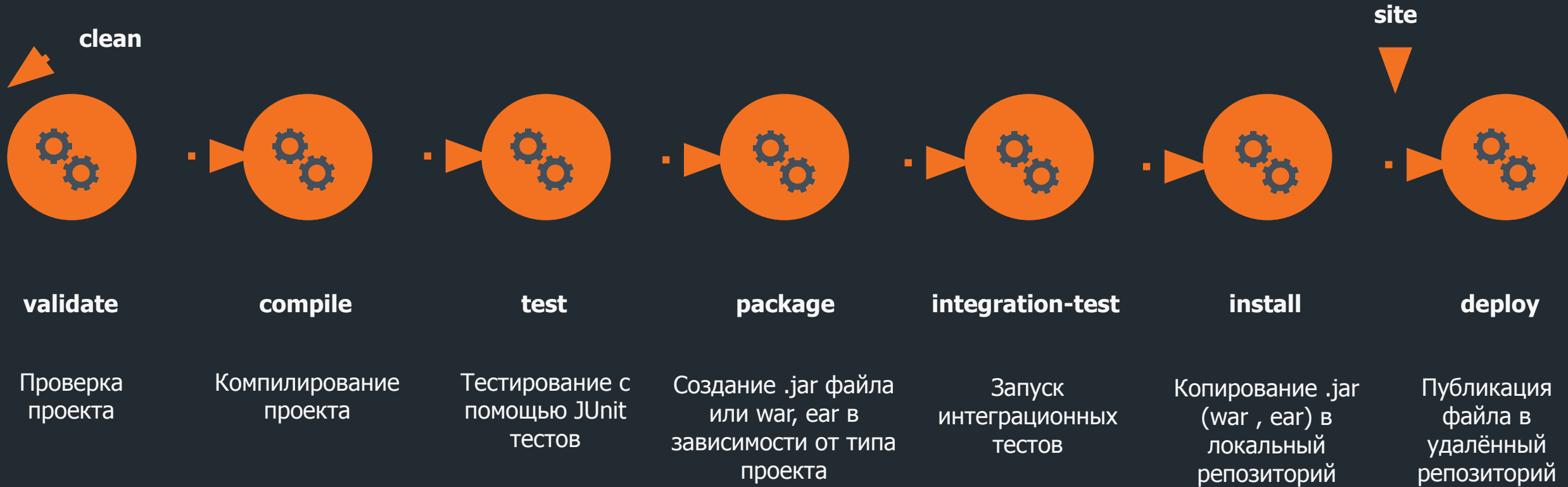
### 6.deploy

Публикация файла в удалённый репозиторий





# Maven



# Maven



pom.xml

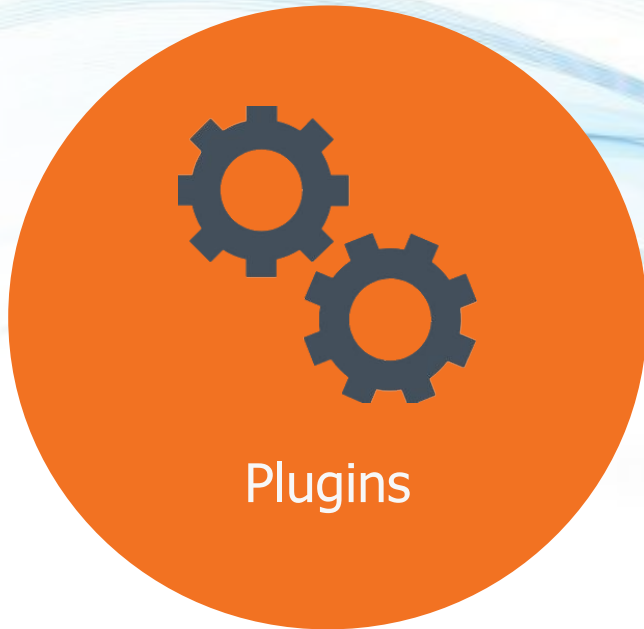
## Область действия зависимости

Область действия зависимости **scope** определяет этап жизненного цикла проекта, в котором эта зависимость будет использоваться. Maven использует 6 областей:

- ✓ **compile** - область по умолчанию;
- ✓ **provided** - очень похоже на compile, но эта зависимость в сборку не попадает;
- ✓ **runtime** - зависимость с данной областью видимости не обязательна для compilation и используется в фазе выполнения;
- ✓ **test** - зависимость используется при тестировании кода приложения;
- ✓ **system** - область похожа на provided за исключением того, что необходимо определить физическое расположение артефакта на диске. Артефакт с этой областью видимости maven не ищет в репозитории;
- ✓ **import** - эта область используется в зависимости секции `<dependencyManagement>` при сложных связях



# Maven Plugins



## Apache Maven Plugins

Плагины - это способ расширить функциональность maven в больших диапазонах.

Как и зависимости плагины идентифицируются с помощью GAV.

```
<plugin>  
  <groupId>groupId</groupId>  
  <artifactId>artifactId</artifactId>  
  <version>version</version>  
</plugin>
```

`<executions>` - привязка к фазам;

`<configuration>` - передача конфигурации.

Формат artifactId:

`maven-<plugin_name>-plugin` - для собственных плагинов;  
`<plugin_name>-maven-plugin` - зарезервировано Apache Maven.

# Maven Plugins

## Apache Maven Plugins

Запуск плагинов:

1. Общий вид:

```
mvn groupId:artifactId:version:goal
```

```
mvn
```

```
by.epam.inttraining2019.lecture3:greeting-maven-plugin:1.0-SNAPSHOT:greeting
```

2. Сокращенный вид:

1) Если в локальном репозитории есть только одна версия плагина, то версию можно опустить.

```
mvn groupId:artifactId:goal
```

```
mvn by.epam.inttraining2019.lecture3:greeting-maven-plugin:greeting
```

2) Если соблюдается соглашение об именовании artifactId плагина, то можно использовать только уникальное имя.

```
mvn greeting:greeting
```



# Maven Plugins

## Apache Maven Profiles

Внутри профилей могут быть определены следующие теги:

- `<repositories>`
- `<pluginRepositories>`
- `<dependencies>`
- `<plugins>`
- `<properties>`
- `<modules>`
- `<reporting>`
- `<dependencyManagement>`
- `<distributionManagement>`
- `<build>`

При активации профиля его содержимое объединяется с общей частью `pom.xml`.

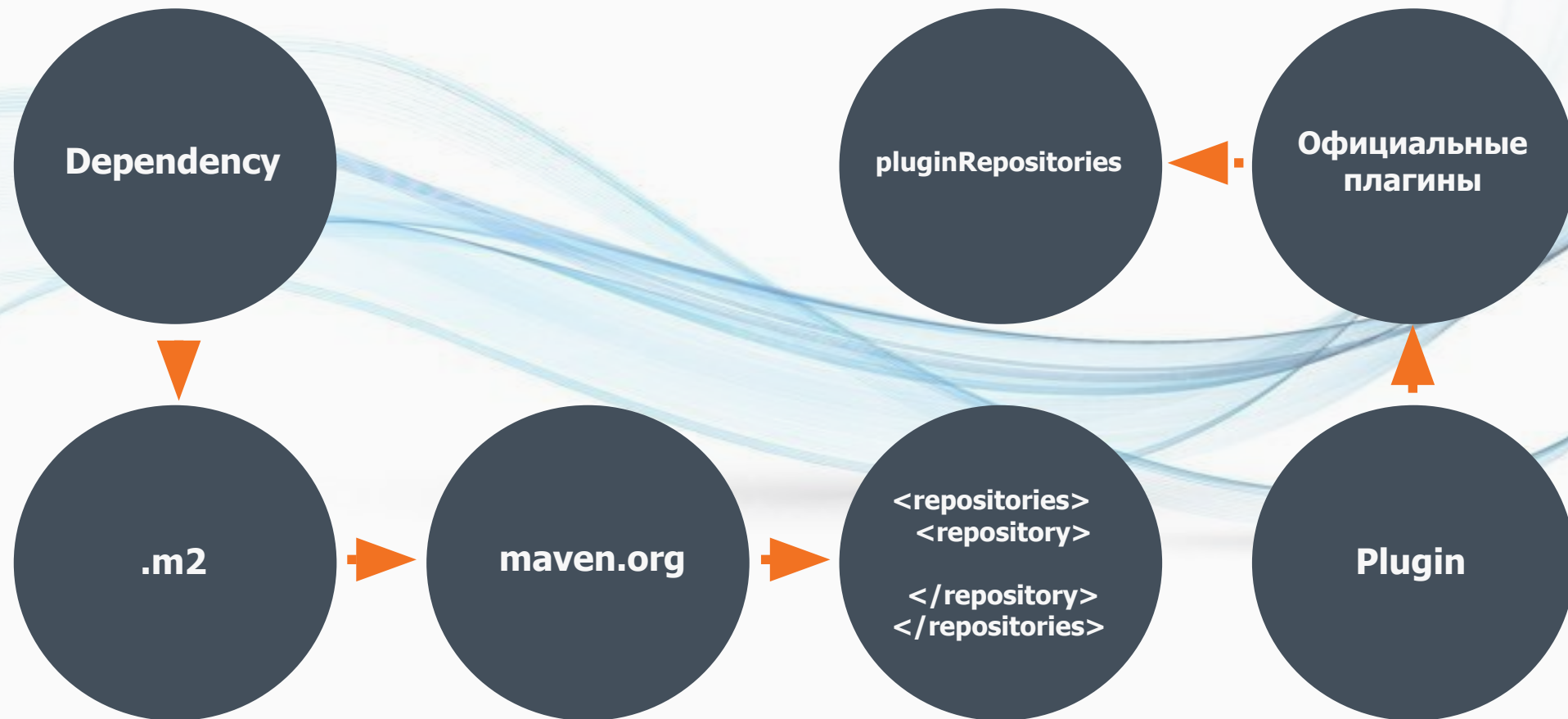
### Активация профилей:

- 1) Вручную – ключ `-P`;
- 2) В файле `pom.xml` - `<activation>`;
- 3) В файле `settings.xml`- `<activeProfiles>`.

**`mvn help:effective-pom -P dev`**



# Maven





<ерат>

# Инструментарий разработчика