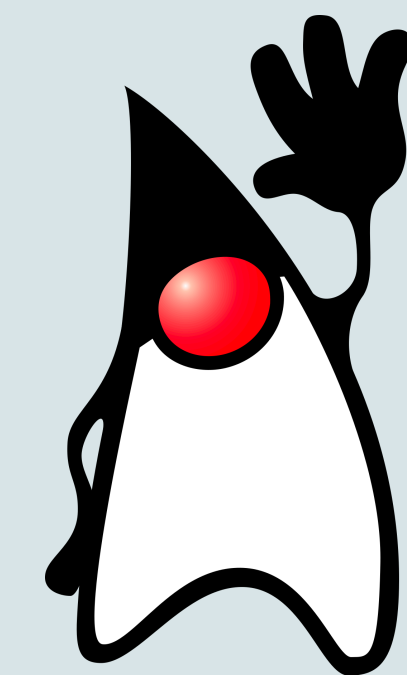




СБЕРБАНК

Корпоративный
университет



DevOps. Рекомендации на собеседовании

Занятие №22

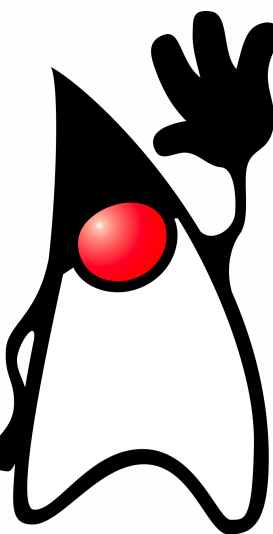


СБЕРБАНК

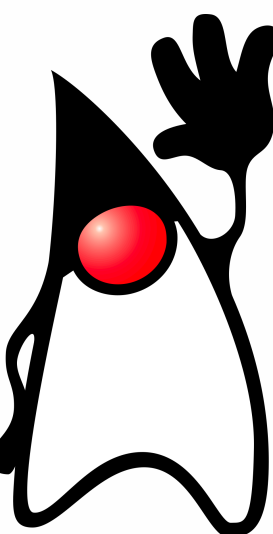
Корпоративный
университет



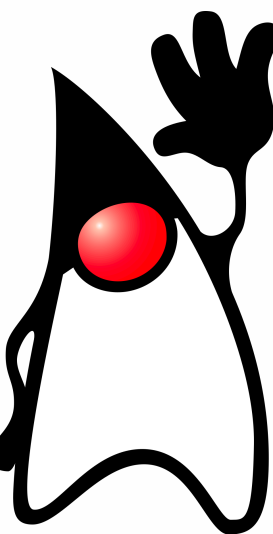
- DevOps
- Вопросы-ответы
- Рекомендации к собеседованию



- Активно участвуем. Не стесняйтесь задавать вопрос.
- Но off-topic обсуждаем в Telegram @sb_ku_java_2019_10
- Не стесняйтесь просто спрашивать в telegram.
-



- Не удалось посмотреть одновременно с занятиями. Не заканчиваем, общаемся в online с теми кто уже начал

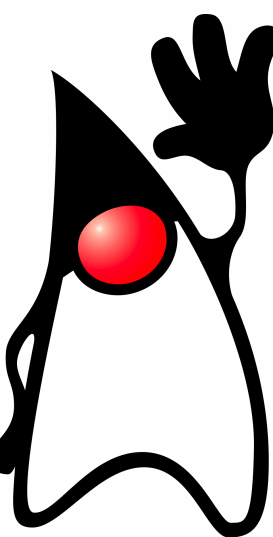


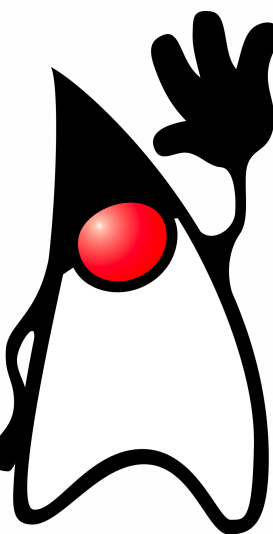
**Договорились?
Поехали!**

01

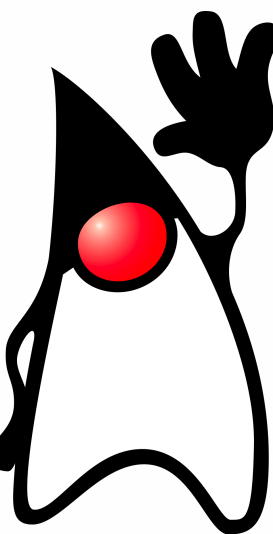
DevOps

- Чем профессионал отличается от любителя?
- Что является итогом работы разработчика?



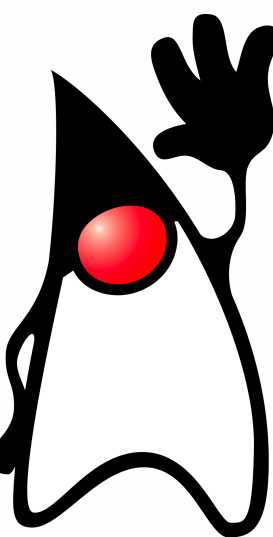


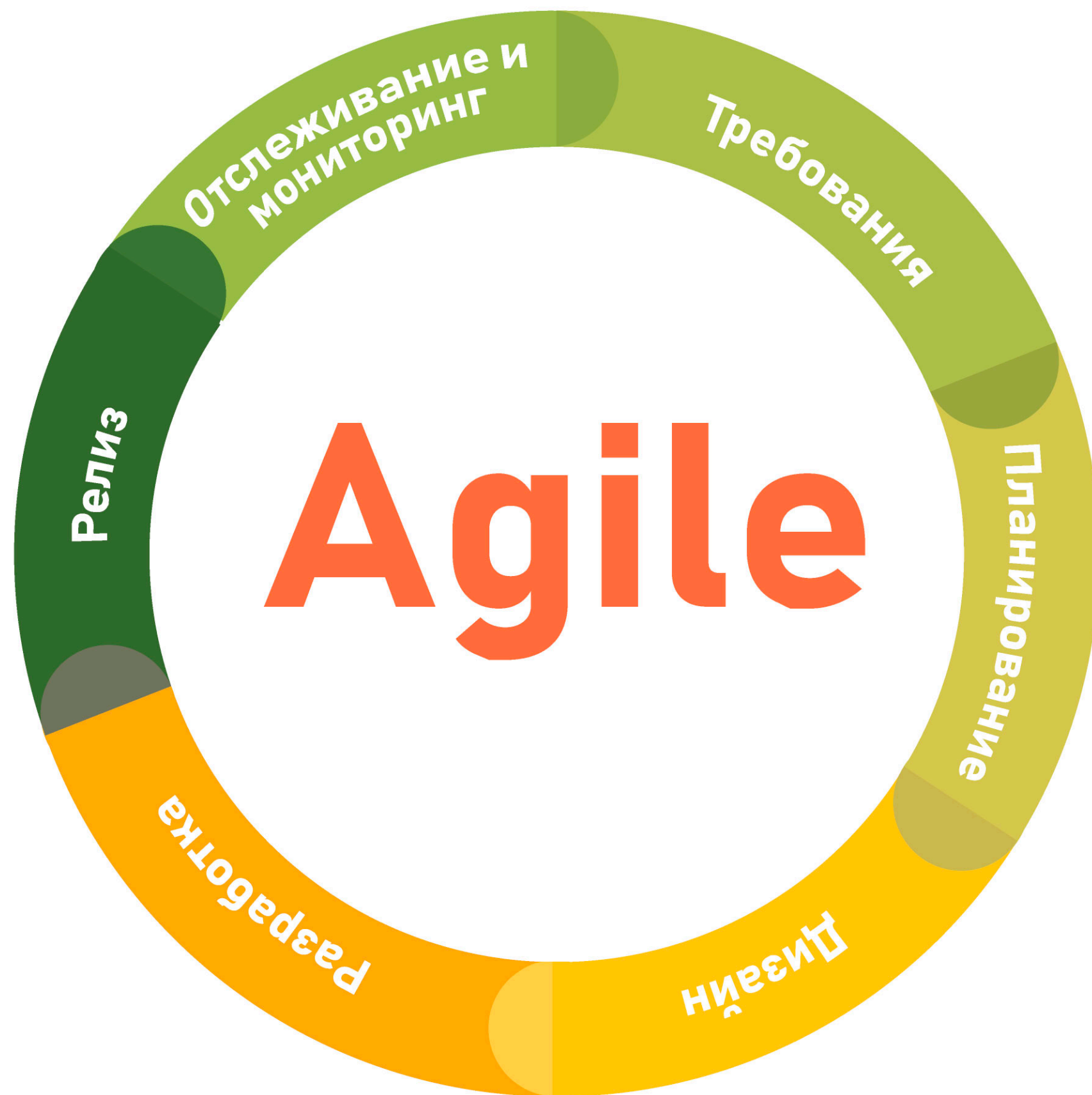
- Code
- Build
- Test
- Package
- Release
- Configuration
- Monitoring



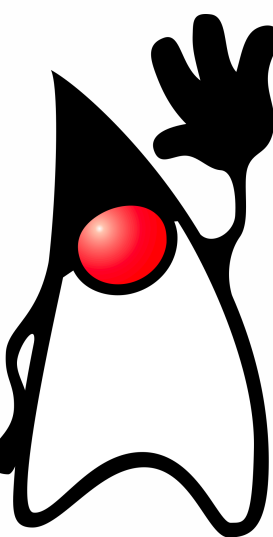
DevOps vs Agile

DevOps vs Continuous Delivery

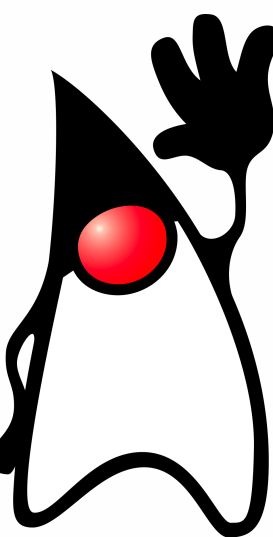




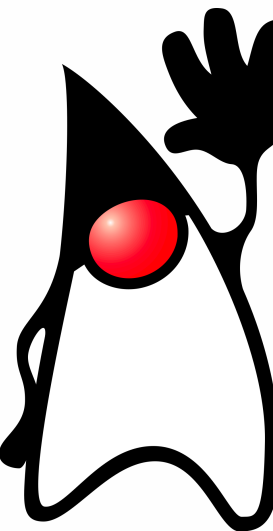
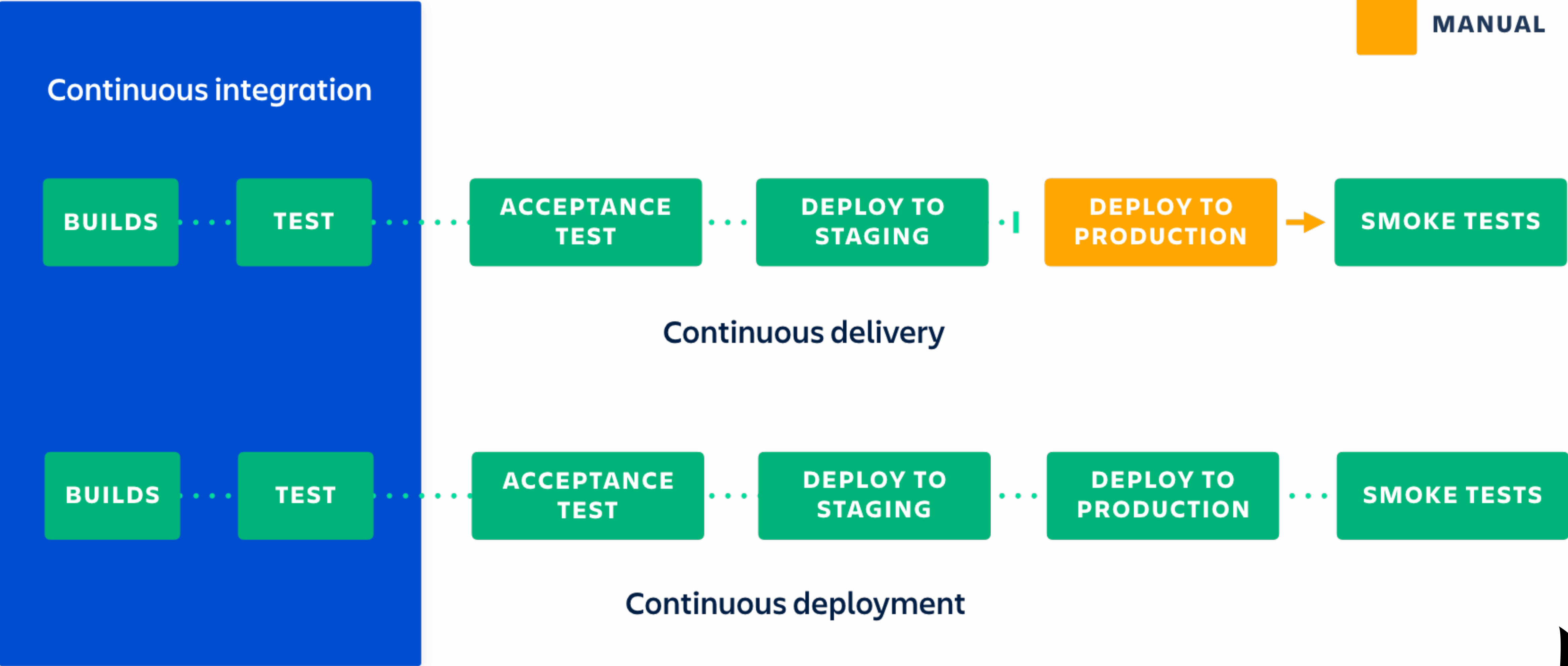
Waterfall

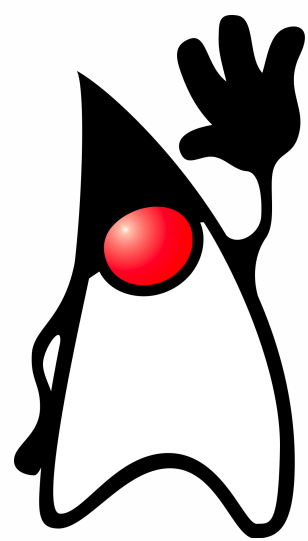
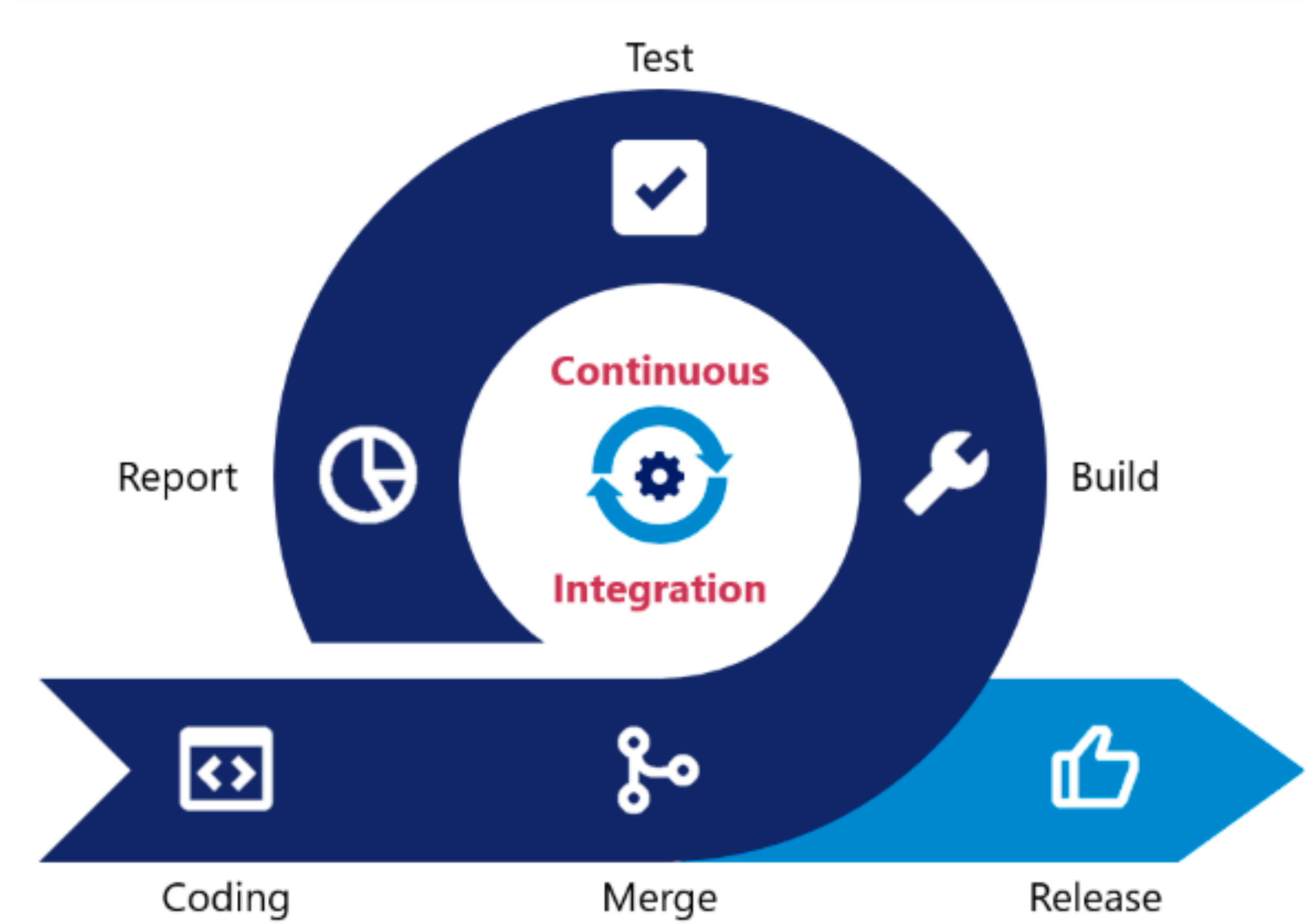


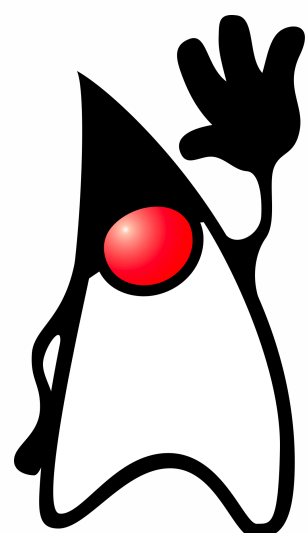
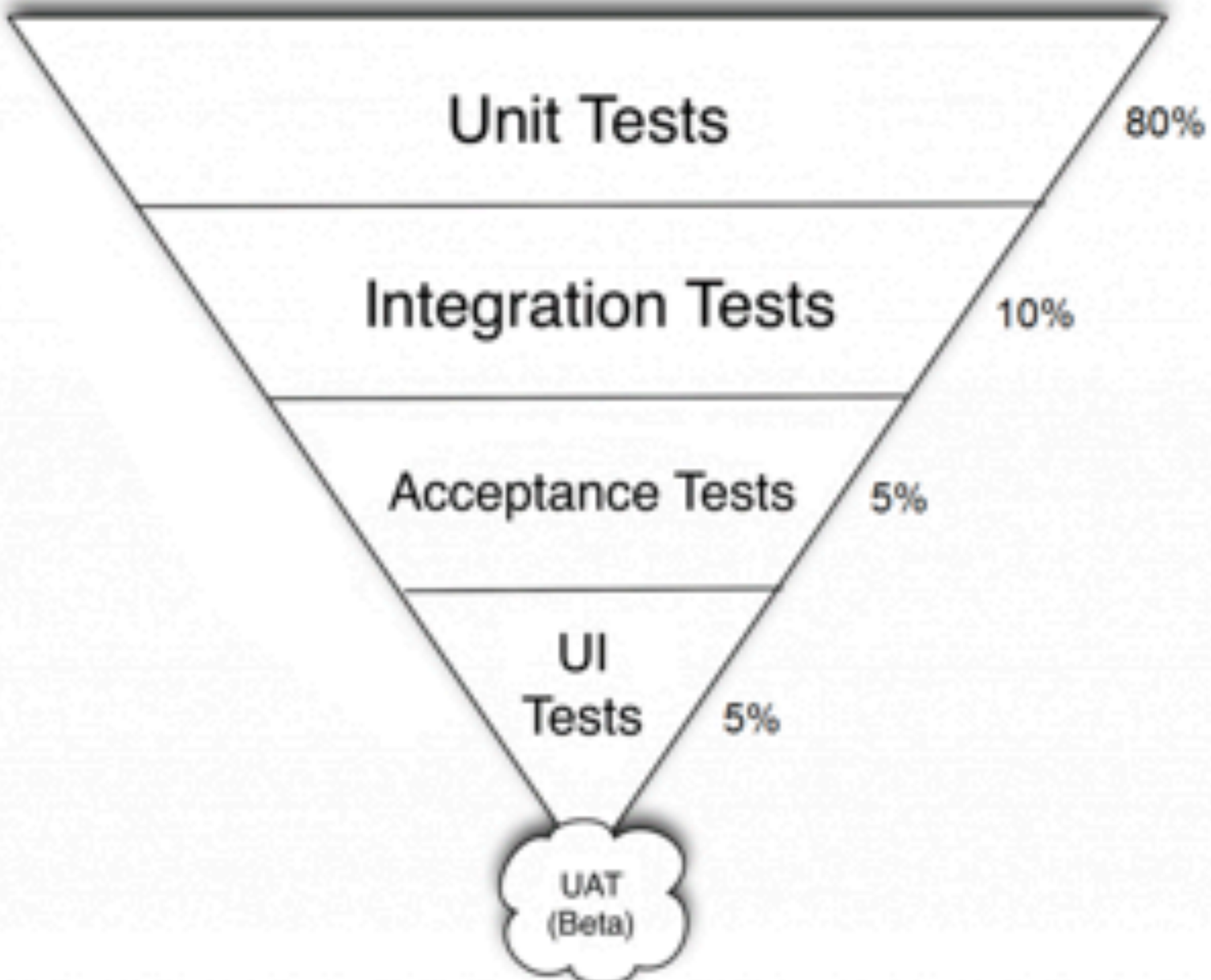
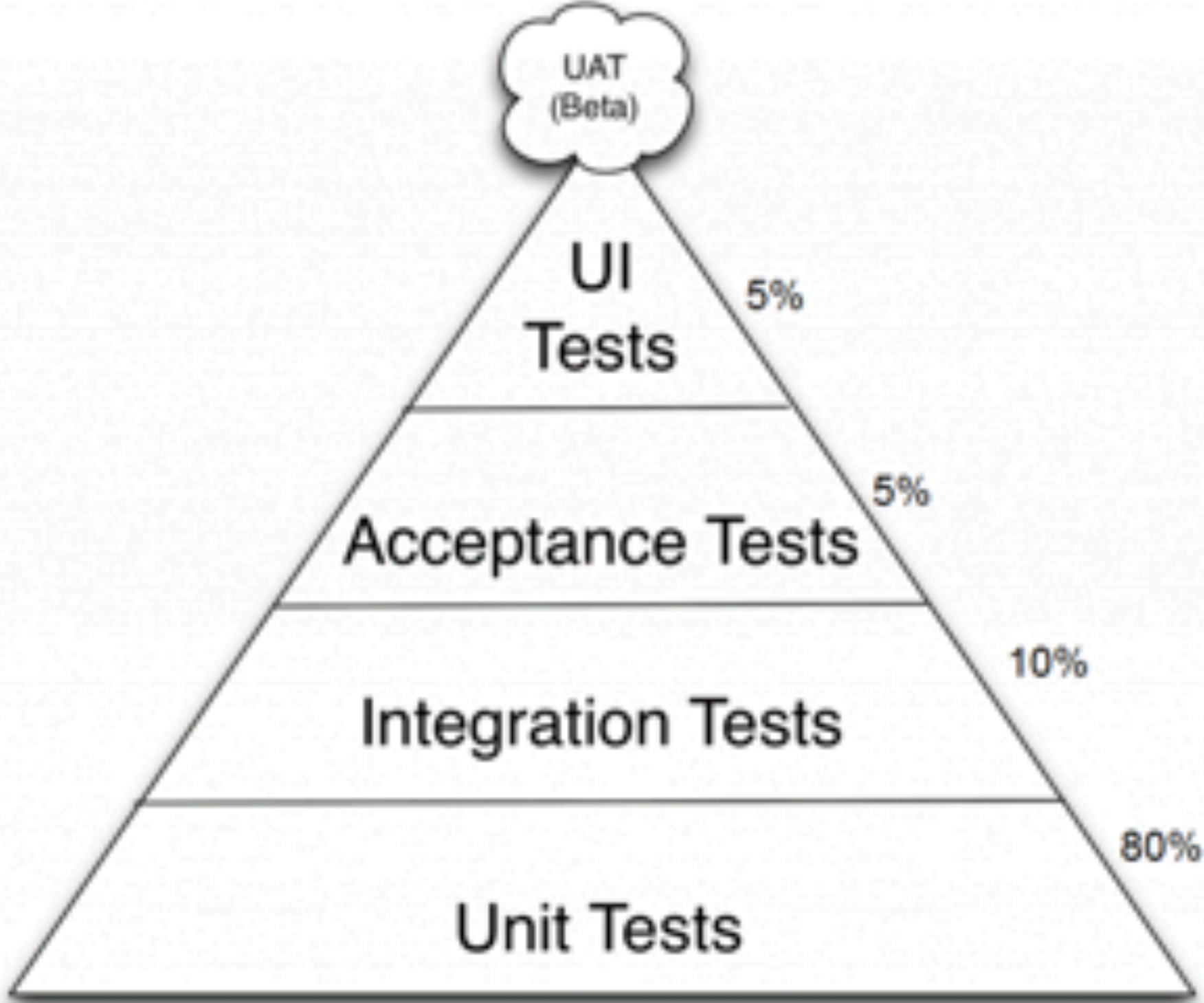
- Сокращение Time-To-Market
- Снижение количества отказов
- Ускорение исправления
- Уменьшения времени на восстановление
- Быстрое реагирование на изменение



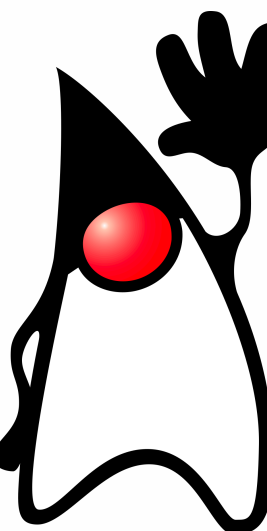
Continuous deployment







Continuous Delivery. Средства

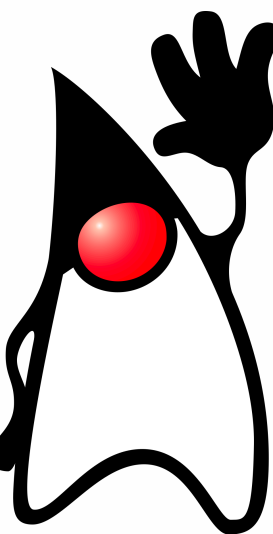


Ваши вопросы?

02

Ответы на вопросы

- Зачем оно?
- Плюсы и минусы?



02.1

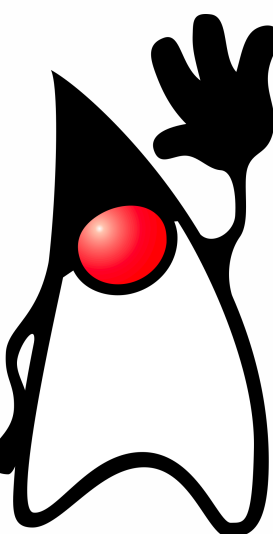
**Функциональное
программирование**

In computer science, functional programming is a programming paradigm—a style of building the structure and elements of computer programs—that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.

Википедия

Два кита:

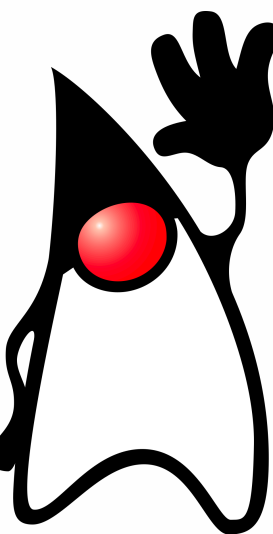
- чистые (детерминированные) функции
- немутлирующие данные.



Чистые (детерминированные) функции
упрощают отладку «в уме»

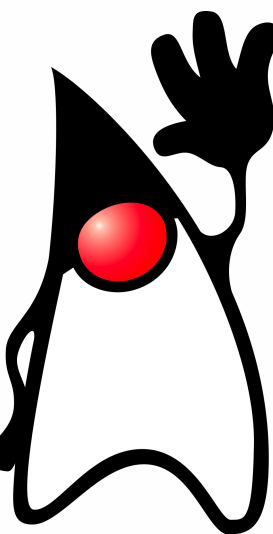
Немутирующие данные значительно упрощают многопоточные
алгоритмы.

Нет общего состояния – нет проблемы
многопоточности.



Пример: ComparatorExample

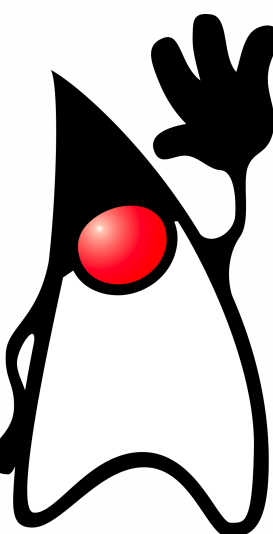
От анонимных классов до лямбд и
функциональных интерфейсов



In computer programming, an anonymous function (function literal, lambda abstraction, or lambda expression) is a function definition that is not bound to an identifier. Anonymous functions are often arguments being passed to higher-order functions, or used for constructing the result of a higher-order function that needs to return a function.

Википедия

higher-order function это функция, которая
в качестве параметров принимает другую функцию
или возвращает функцию как результат.



В Java лямбда функции реализуются,
через функциональные интерфейсы:

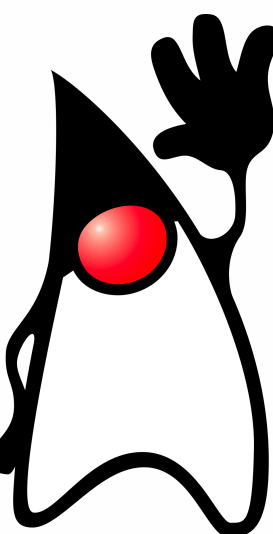
Примеры функциональных интерфейсов:

`java.util.function.Supplier`

`java.util.function.Predicate`

`java.util.function.Function`

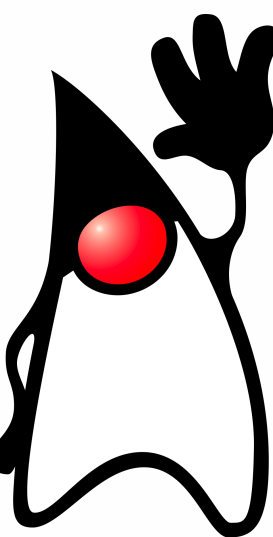
Пример: Lambda



Основное правило.

Никакого изменения объектов. Только новые.

Пример MutationExamples



Формальное определение очень уж тяжело, для начала можно глянуть объяснение [тут](#).

Грубое определение.

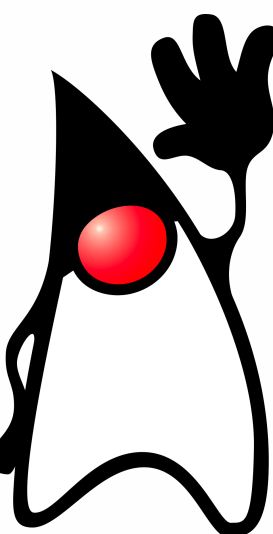
Монада – это контейнер с результатом операции и функции для обработки этого результата.

Из них можно удобно конструировать цепочки преобразований входных данных.

Maybe monad – это `java.util.Optional`

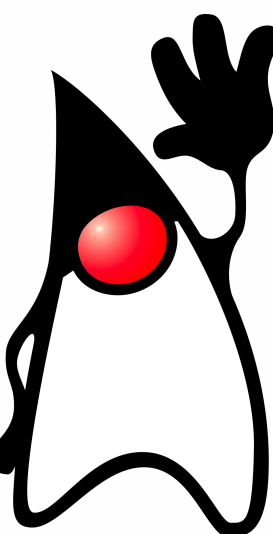
Пример MonadExample

[Пример с библиотекой vavr](#).



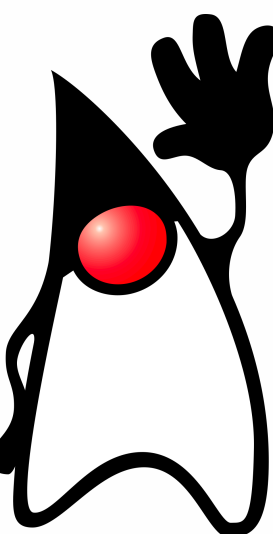
Stream – это последовательность данных и декларативное описание последовательности действий, которые надо выполнить над этими данными

Примеры: `package stream`



Ваши вопросы?

1. Можно ли как-то получить скидку на курсы или полезное видео с сайта <https://otus.ru/> ?
2. Принципиальные отличия war и jar, когда мы прописываем в pom.xml в packaging
3. При работе с числами/деньгами как правильно округлять значения?
4. В jsr файлах, когда мы делали библиотеку не совсем понятна 'магия' как правильно выводить данные/значения из базы данных. Полезные ссылки и дом материал с правилами нахватает.
5. Презентация 20 стр 57. «между сервисами-бинами» освежите в памяти это место.

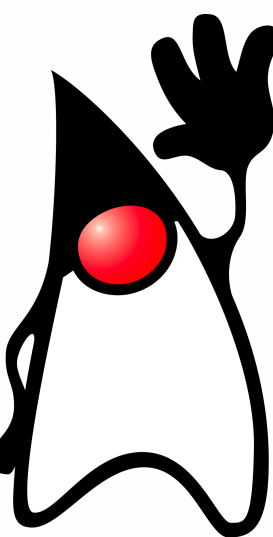


Ваши вопросы?

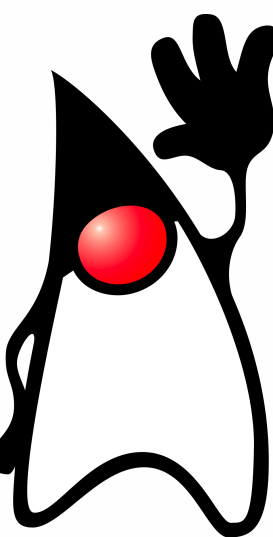
03

Подготовка к собеседованию

- Модификаторы доступа и иерархия классов-пакетов, области видимости переменных?
- Quizful, proghub... Пробное тестирование
- Потоки, мониторинг, volatile, happens before
- Коллекции, иерархия коллекций, LinkedList vs ArrayList. Устройство HashMap, сложности алгоритмов
- Сериализация (xml, json)
- БД. 1, 2, 3 нормальные формы, уровни изоляций транзакций

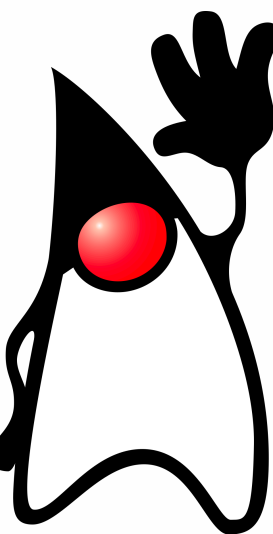


- Базовые паттерны
- IoC, Spring IoC, скопы для бинов
- Функциональное программирование в Java
-



Что дальше?

- Стандартный тест по Java в банке.
- Передача вас HR с моими отзывами



Ваши вопросы?

Спасибо за внимание!

